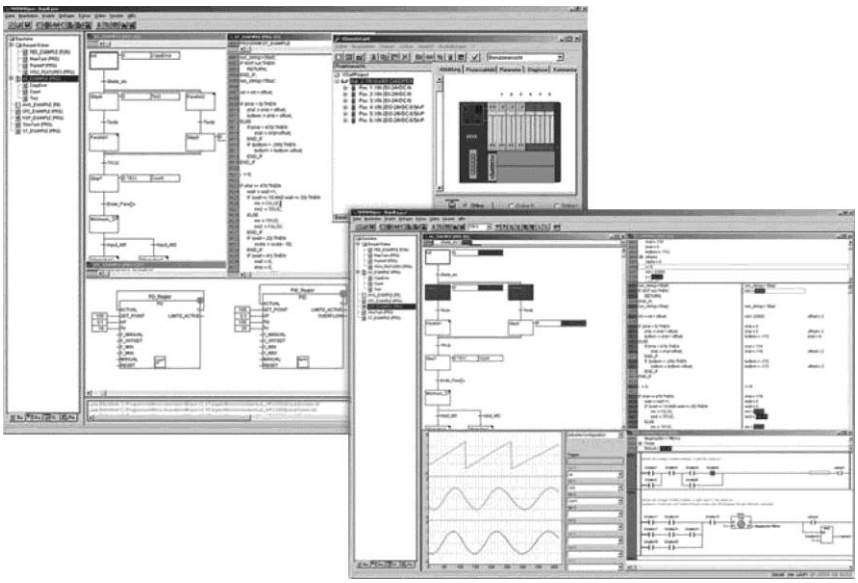
PLC programming XC152



Imprint

Manufacturer

Eaton Automation GmbH
Spinnereistrasse 8-14
CH-9008 St. Gallen
Switzerland
www.eaton-automation.com
www.eaton.com

Support

Region North America
Eaton Corporation
Electrical Sector
1111 Superior Ave.
Cleveland, OH 44114
United States
877-ETN-CARE (877-386-2273)
www.eaton.com

Other regions
Please contact your local distributor or send an e-mail to:
automation@eaton.com

Original instructions

German

Editor

Daniel Lenherr

Brand and product names

All brand and product names are trademarks or registered trademarks of the owner concerned.

Copyright

© Eaton Automation GmbH, CH-9008 St. Gallen

All rights reserved, also for the translation.

None of this documents may be reproduced or processed, duplicated or distributed by electronic systems in any form (print, photocopy, microfilm or any other process) without the written permission of Eaton Automation GmbH, St. Gallen.

Subject to alteration

Contents

1 General 6
1.1 Purpose of this document 6
1.2 Comments about this document 6
1.3 Additional documentation 6

2 Install 7
2.1 Equipment supplied 7
2.2 System requirements 7
2.3 Install PLC programming tool 8
2.4 Install PLC target systems 9

3 Uninstall 10
3.1 Uninstall PLC programming tool 10
3.2 Uninstall PLC target systems 10

4 Target settings 11
4.1 Target platform 11
4.2 Memory layout 12
4.2.1 Overview memory layout 13
4.3 General 14
4.4 Network functionality 15
4.5 Visualization 16

5 PLC configuration 17
5.1 Working in the PLC configuration 17
5.1.1 Project specific configuration files and device files 19
5.2 General settings 20
5.3 Configuration as CAN-Master 21
5.3.1 Configuration of CAN-Master 22
5.3.2 Configuration of CAN-Nodes 25
5.3.3 Bus diagnostic 31
5.4 Configuration as CAN-Device 32
5.4.1 Configuration of CAN-Device 33
5.4.2 Configuration of CAN-Device in CAN-Master 37
5.5 Configuration as Profibus-Master 38
5.5.1 Configuration of Profibus-Master 39
5.5.2 Configuration of Profibus-Slaves 44
5.5.3 Bus diagnostic 50
5.6 SmartWire-DT configuration 51
5.6.1 Configuration of the SWD master and SWD module selection 52
5.6.2 Bus diagnostic 56
5.6.3 Acyclic Communication 56
5.6.4 SmartWire-DT LEDs 56
5.6.5 Config button 57
5.6.6 Browser commands 58

6 Operation 59

Contents

6.1	Startup behaviour	59
6.2	Switch off behaviour	59
6.3	Operating state of controller	60
6.3.1	Overview operating- and error states	61
6.4	Switching the operating state	62
6.5	Start behaviour	63
6.6	Stop behaviour	64
6.7	Reset behaviour	64
6.8	Test and startup	65
6.9	Program transfer	66
6.9.1	Create boot project	67
7	**Program execution and system time**	**68**
7.1	Program execution	68
7.2	Task configuration	69
7.3	Multitasking	71
7.4	Task monitoring / Watchdog timing	71
7.5	Data retention	74
7.6	Direct pheripherie access	74
7.7	Interrupt processing	74
7.8	System libraries, function blocks and functions	75
7.9	Process image / IO-Update	75
7.9.1	CAN-Bus	76
7.9.2	Profibus	76
7.9.3	SmartWire-DT	76
8	**Connection establishment programming PC – Controller**	**77**
8.1	Connection establishment with ethernet	77
8.2	Scan/Modify the IP address	80
9	**Parameter manager / Object directory**	**81**
10	**PLC browser**	**82**
11	**Alarm configuration**	**83**
12	**Connecting to visualization / Generating of the symbol file**	**84**
12.1	Configure symbol file	84
12.2	Download symbol file	87
13	**Target system installation and firmware update**	**88**
14	**Licensing**	**90**
14.1	PLC programing tool	90
14.2	PLC runtime system	90
14.3	Target visualization	90
14.4	Web visualization	90
15	**Alteration index**	**91**

Contents

1 General

1.1 Purpose of this document

This document describes the use of the PLC programming tool "XSOFT-CODESYS-2" and the PLC runtime system for XC152 device type with Windows CE. This document serves as addition of the user manual PLC programming tool CoDeSys V2.3 of the company 3S-Smart Software Solutions GmbH.

→ Dialogs and examples in this document are standardized. Depending on selection of the controller type therefore dialogs can differ.

1.2 Comments about this document

Please send any comments, recommendations or suggestions relating to this document to automation@eaton.com.

1.3 Additional documentation

The following documents may be helpful in the use of the device in addition to this document.

The following documents are available in the form of PDF file for download from our home page (www.eaton-automation.com):

[1] MN05010007Z-EN
System description Windows CE

[2] Various
Documentations concerning PLC function libraries
MN04802103Z-EN
SWD_Util library

[3] -
System description CiA Draft Standard DSP301

[4] -
User manual PLC programming tool CoDeSys V2.3

[5] MN05006002Z-EN
Manual SmartWire-DT System

[6] MN05013002Z-EN
Manual SmartWire-DT Gateways

[7] MN05006001Z-EN
Manual SmartWire-DT Units

2 Install

The "XSOFT-CODESYS-2" product is an independent software package. It consists of a software component (PLC programming tool with appropriate PLC target systems) which is installed on any programming PC, and a software component which is installed on PLC target systems with Windows CE (➔ Chap. 14) provided this has sufficient license points.

➔ If you have any questions on license products, please contact your local Eaton sales distributor.

➔ On the controller, a PLC runtime system is already installed. For a necessary update of the operating and/or PLC runtime system see chapter "Target system installation and firmware update" (➔ Chap. 13).

2.1 Equipment supplied

Designation	Version
Software "XSOFT-CODESYS-2" incl. user manual	2.3.9 SP4

Consisting of:

Designation	Version
CoDeSys	2.3.9.44
CoDeSys Gateway	2.3.9.31
CoDeSys OPC-Server [1]	2.3.13.11

PLC target system	PLC runtime system	Operating system (OS)
XC-152-V2.3.9 SP4	PLCWinCE V 2.4.13 (xxx)	Windows CE 5.0 Image Release 2.27.0 (xxx)

2.2 System requirements

Operating system
Windows XP, Windows Vista, Windows 7

[1] Optionale Installation, Komponente ist kein Bestandteil dieser Dokumentation

2.3 Install PLC programming tool

Insert the installation CD into your PC and start the installation with 'Setup.exe'.

→ If no series number or license key is available by the installation of the PLC programming tool, the target systems are installed in the demo mode.

The installation will create the following directories by default:

Directory	Description
Programming system directory C:\Program Files\Eaton\XSOFT-CODESYS V2.3.9 SP4	PLC programming tool
Target system directory C:\Program Files\Common Files\CAA-Targets\Eaton Automation\V2.3.9 SP4	Target specific components such as libraries and configuration data incl. firmware for PLC target systems

Behaviour in relation to installed PLC programming tool

Already installed versions of PLC programming tool "XSOFT-CODESYS-2" and appropriate PLC target systems are not affected.

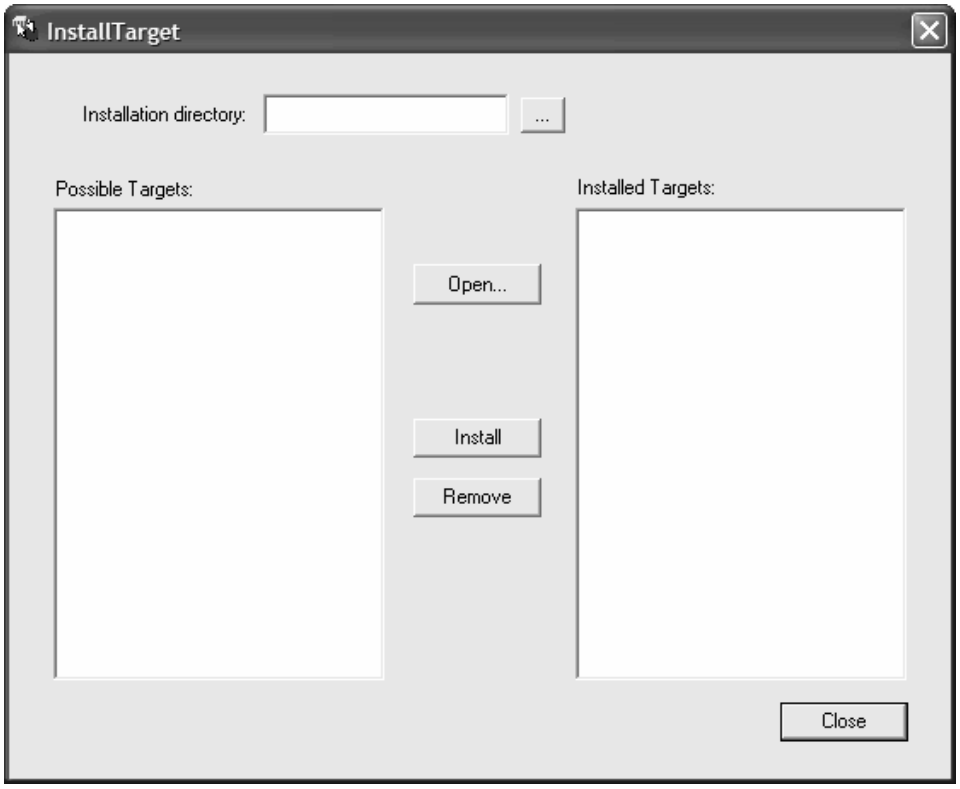
→ All PLC target systems appropriate for installation are assigned a new designation or identification.

→ If you would like to update PLC projects, which are created with older version of the PLC programming tool, then the PLC target system must be changed. After closing and renewed opening of the PLC project, the PLC project will updated and the new libraries merged.

2.4 Install PLC target systems

The PLC target systems appropriate for the PLC programming tool are installed by default with the setup.

However, it is also possible to install or remove PLC target systems at a later time using the 'InstallTarget' function.



3 Uninstall

3.1 Uninstall PLC programming tool

When the PLC programming tool is uninstalled, only installed files and components are removed. Files and directories in the programming system directory must therefore be removed manually if necessary.

3.2 Uninstall PLC target systems

When the PLC programming tool is uninstalled, the target specific components such as libraries and configuration data in the target system directory are not removed.

The installed PLC target systems must be removed using appropriate uninstall routine.

→ Alternatively installed PLC target systems can also be removed from the PLC programming tool using the 'InstallTarget' function.

Subsequently, the target specific components in the target system directory must be removed manually.

4 Target settings

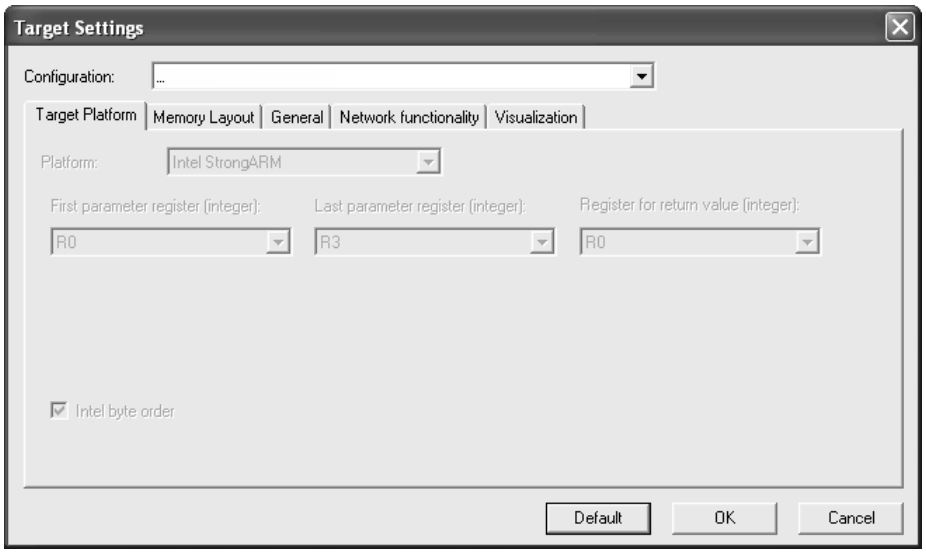
This dialog opens automatically if a new project is created. Otherwise it is reached via <Target settings> in the 'Resources' tab.

In the target settings, select the appropriate PLC type. This configuration selects optimum settings for processor type and memory size. The entry **'None'** automatically activates Simulation mode.

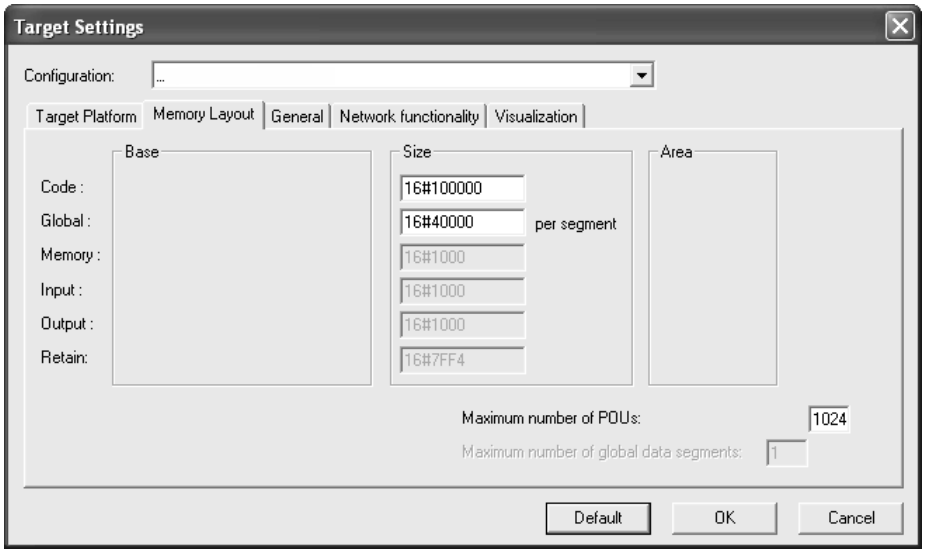
Changes to the preset target settings can have an effect on the behaviour of the target system!

If necessary the **Default** button resets the changed configuration to the standard configuration.

4.1 Target platform



4.2 Memory layout



Code size

Default 1024KB (16#100000) memory for program code. This memory is allocated automatically.

Global size

Default 256KB (16#40000) memory for global data. This memory is allocated automatically.

Maximum number of POUs

Each POU needs 12Byte memory automatically. Altogether 12KB memory is needed for the function pointer table (1024 components -> 12KB). This memory is added to the PLC data memory.

→ The calculation of the number of POUs used in the program includes all functions and function blocks of inserted libraries.

4.2.1 Overview memory layout

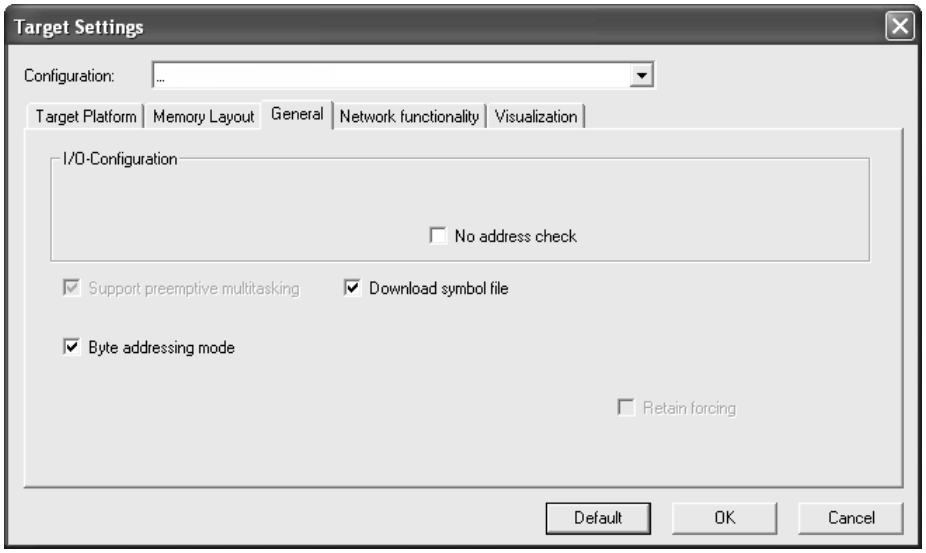
Default settings with online change
Number of data segments = 1
Size of data segment = 256KB

32KB retentive data area
12KB Function pointer table
12KB In/Output marker data
1 * 256KB global data storage area
1024KB program code (Online Change)
1024kB program code

→ During the standard compilation of the PLC program the PLC programming tool displays the size of the data area to the user.

The PLC browser and the command 'sysinfo' can be used to determine the size of the PLC program online (➔ Chap. 10)

4.3 General



No address check

If this option is activated, the addresses are not checked during the compilation.

Download symbol file

If this option is activated, a symbol file will be created during the compilation and will be downloaded

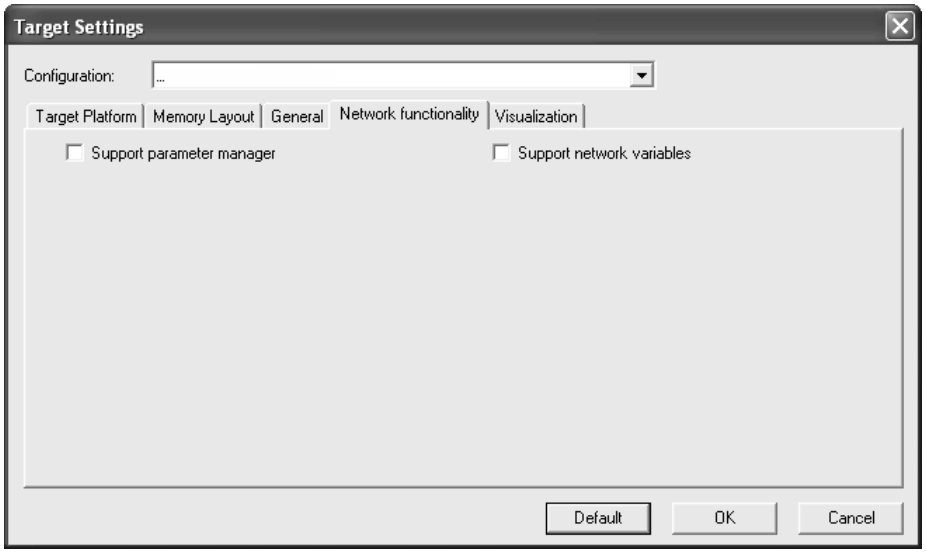
Support preemptive Multitasking

Multitasking is supported by default (cannot changed by user)

Byte addressing mode

If this option is activated, the addressing takes place bytewise (e.g.: %QD4 corresponds to %QB4).

4.4 Network functionality



Support parameter manager

If this option is activated, the entry **Parameter Manager** appears in the 'Resources' tab. This makes it possible to create an object directory for variables and parameters, which are used for a controlled and active data exchange with other PLCs.

→ The object directory functionality is supported in conjunction with an inserted CAN-Device in the PLC configuration.

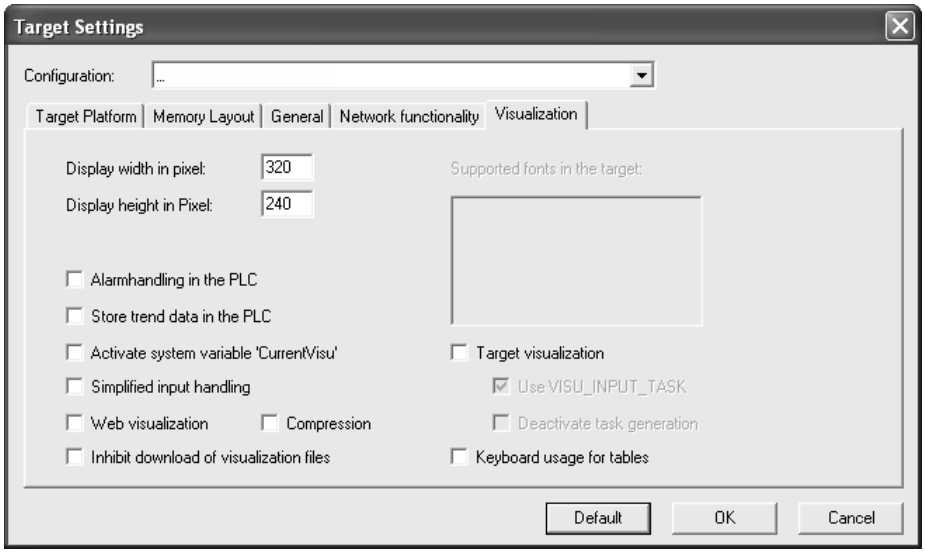
→ Please refer to detailed information in the CoDeSys V2.3 user manual or the online help of the PLC programming tool.

Support network variables

If this option is activated, the network variables can be used. They are used for automatic data exchange with other PLCs.

→ Please refer to detailed information in the CoDeSys V2.3 user manual or the online help of the PLC programming tool.

4.5 Visualization



Depending on the used PLC target system, Target visualization and Web visualization are supported. Appropriately information is given in the table below.

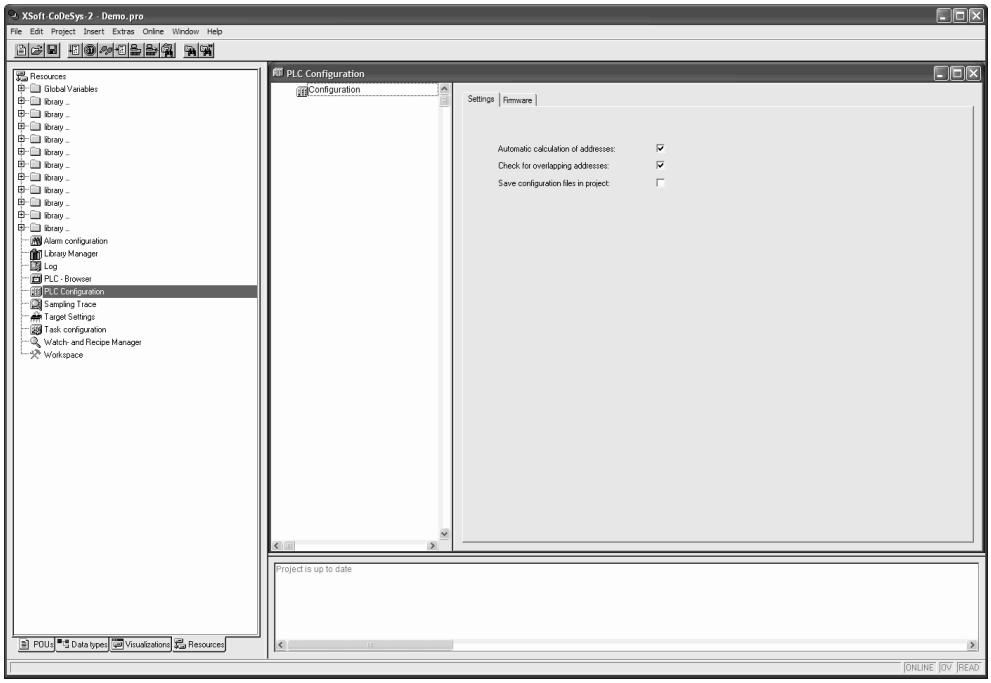
PLC target system	PLC runtime system	Target-Visu	Web-Visu
≥ XC-152-V2.3.9 SP3	≥ PLCWinCE V 2.4.10 (xxx)	✓	✓

→ Please refer to detailed information in the CoDeSys V2.3 user manual, the CoDeSys visualization user manual or the online help of the PLC programming tool.

5 PLC configuration

The PLC configuration is found as an object on the 'Resources' tab in the 'Object Organizer'.

The PLC configuration enables inputs and outputs to be configured or bus-compatible I/O modules to be connected. Configuration files (*.cfg) and device files (e.g. *.gsd, *.eds) are used as the basis for work in the PLC configuration.



The PLC configuration is displayed in the editor in tree structure and can be edited using menu commands and dialogs. The configuration contains input and/or output elements and also management elements which themselves also have subelements (for example, CAN, PROFIBUS).

Input and output appear in the editor with the IEC address by which they can be accessed. Each input and output can be tagged with a symbolic name which is located before IEC address.

5.1 Working in the PLC configuration

The PLC configuration editor is divided up in two parts. In the left window the configuration tree is displayed. In the right window the currently available configuration dialogs are shown on one or several tabs.

At the top of the configuration tree the entry of the "root" module is shown with a name which has been defined in the configuration file *.cfg. Below this are shown hierarchically indented the other elements of the configuration: Modules of different types (CAN, PROFIBUS, I/O), channels or bit channels.

Selecting elements

In order to select elements, click the mouse on the corresponding element, or use the arrow keys to move the dotted rectangle onto the desired element.

Elements that begin with a plus sign are organization elements and contain subelements. To open an element, select the element and double-click the plus sign or press <Enter>. You can close opened elements (minus sign in front of the element) the same way.

Inserting elements, <Insert> <Insert element>, <Insert> <Append subelement>

Depending on the definitions in the configuration file(s) and on the available device files, which have been read when the project was opened, a basic composition of elements is automatically positioned in the configuration tree. If one of those elements is selected, further elements may be added if this is allowed by the definitions in the configuration file and if the needed device files are available:

Menu item <Insert> <Insert element>: An element can be selected and inserted before the element which is currently marked in the configuration tree.

Menu item <Insert> <Append subelement>**:** An element can be selected and inserted as subelement of the element which is currently marked in the configuration tree. It will be inserted at the last position.

The most important commands are found in the context menu (right mouse button).

Replacing/switching elements, <Extras> <Replace element>

Depending on the definition in the configuration file, the currently selected element can be replaced by another. It is also possible to switch channels, which are set up in a way that they can be used as input or as output elements. Use the menu item <Extras> <Replace element>.

Recalculation of Module addresses, <Extras> <Calculate addresses>

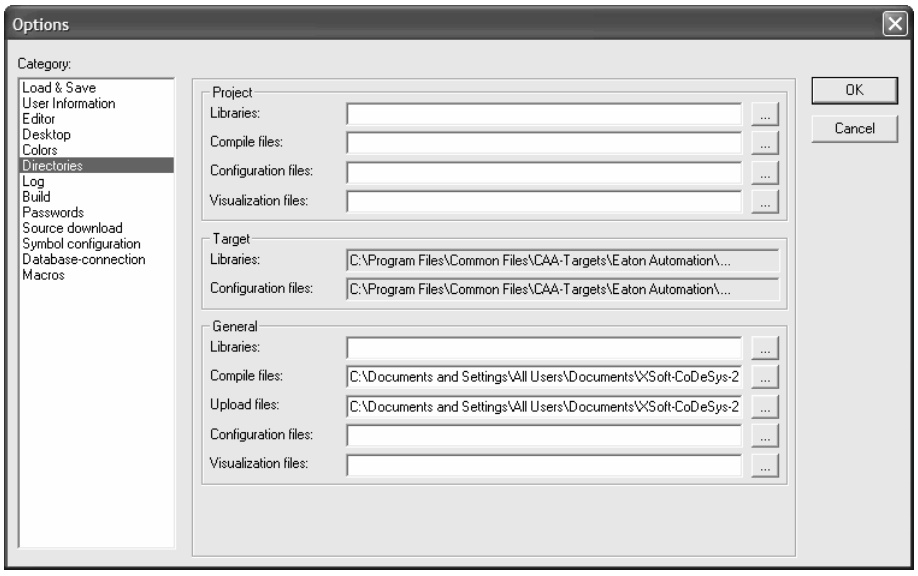
If the option **Calculate addresses** is activated in the dialog 'Settings' of the PLC configuration editor, then the command 'Extras', 'Calculate addresses' will start to recalculate the addresses of the modules. All modules starting with the one, which is currently selected in the configuration tree, will be regarded.

Return to standard configuration, <Extras> <Standard configuration>

The command 'Extras', 'Standard configuration' can be used to restore the original PLC configuration, which is defined by the configuration file *.cfg of the target system.

5.1.1 Project specific configuration files and device files

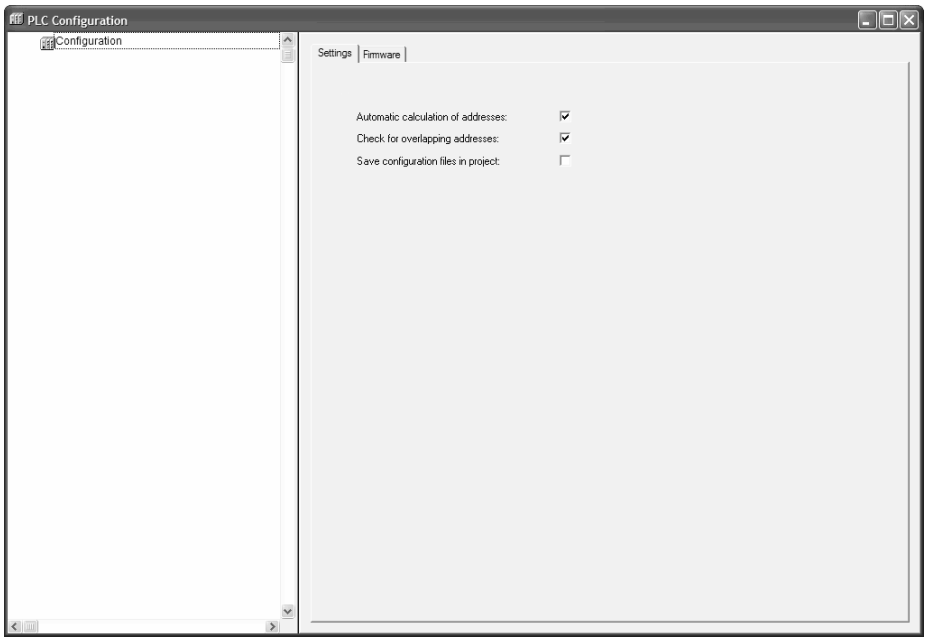
Customized directories for configuration file(s) and device file(s) can be defined project specific with menu item <Project> <Options>. In the project specific library and configuration path, relative directory paths to the project can be defined (e.g. *.\Libraries, .\PLCconf*).



→ The project must be closed after defining project specific directories for the configuration file(s) and device file(s).

After subsequent opening of project the additional configuration file(s) and device file(s) are visible in the PLC configuration and can be inserted.

5.2 General settings



Automatic calculation of addresses

If this option is activated, each newly inserted module is automatically allocated with an address, which the result of the address of the module inserted beforehand plus the size of this address. If a module is removed from the configuration, the addresses of the subsequent modules are adjusted automatically. When the command 'Extras', 'Calculate addresses' is executed, all addresses starting at the selected node (module) will be recalculated.

Check for overlapping addresses

If this option is activated, the project will be checked for overlapping addresses during compilation and a corresponding message will be displayed

Save configuration files in project

If this option is activated, the information which is contained in the configuration file(s) and the device description files will be saved in the project.

5.3 Configuration as CAN-Master

The PLC programming tool supports a hardware configuration according to the CANopen Draft Standard 301. This requires a configuration file, which allows CAN modules to be inserted.

All EDS files (**E**lectronic **D**ata **S**heet) and DCF files (**D**evice **C**onfiguration **F**ile) which are stored in the defined configuration files directory, can be integrated, edited and displayed in the PLC configuration. In the EDS file the configuration options of a CAN module are described. If you add a module which is described in a DCF file, only the IEC addresses can be modified.

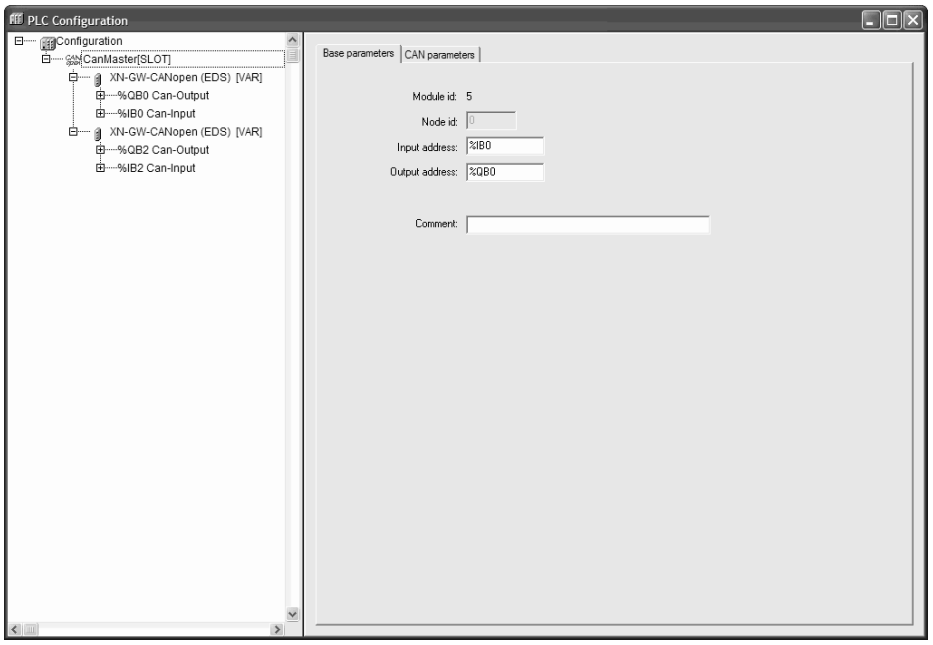
The modules receive a configuration, which describes the timing and error behaviour of the data transmission. Furthermore, the mapping of the PDOs (Process Data Objects) is specified for each module, which is used for sending and receiving (Receive PDO Mapping or Send PDO Mapping dialogs). The values of the available SDOs (Service Data Objects) can be changed (Service Data Objects dialog).

Additional parameters of a CAN module or a CAN Master can be configured in the dialog parameter.

→ In the PLC configuration defined PDOs, which have inputs or outputs that are not used in the PLC program, are not updated by default in the process image.

5.3.1 Configuration of CAN-Master

5.3.1.1 Base parameters



Node number

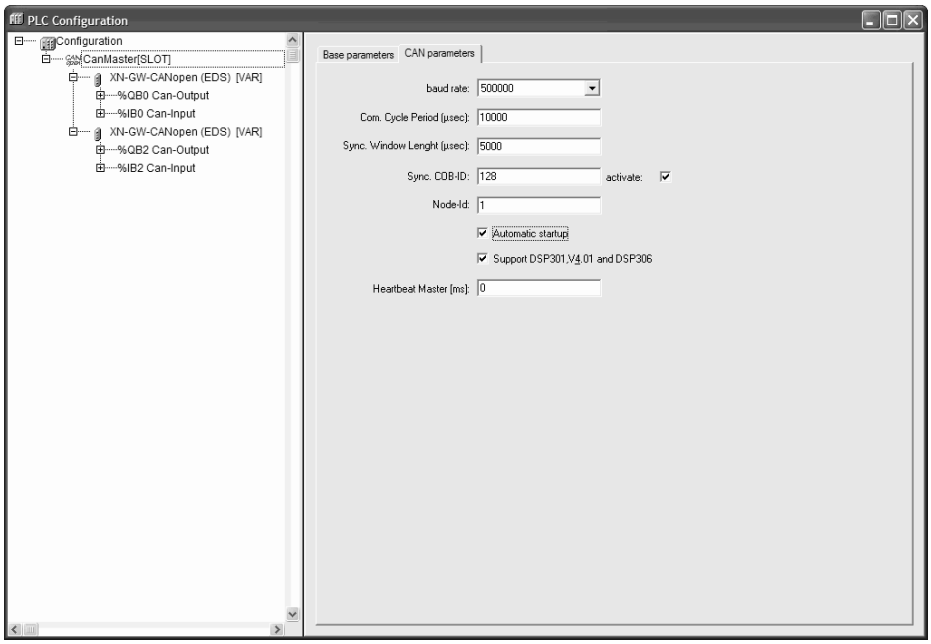
The node number is defined by an entry in the configuration file or - if there is no entry - by the position of the module in the configuration structure and is not editable (not to be confused with the node-Id, which can be defined by the user).

Input address, Output address

This contains the IEC addresses starting from which the PDOs (Process Data Object) in the project can be addressed. It depends on the general settings and the definitions in the configuration file, which addresses are already predefined, which address mode is valid and whether the addresses can still be edited here.

5.3.1.2 CAN parameters

In this dialog the global settings and monitoring parameters for the CAN bus are defined.



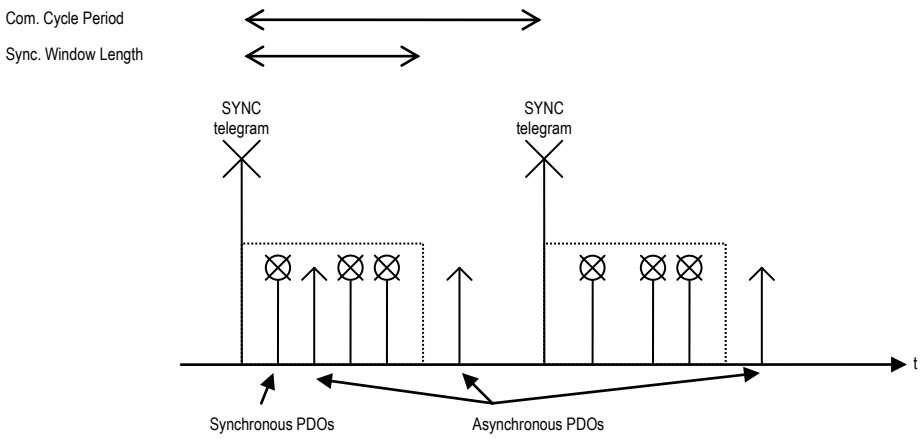
Baud rate

Baud rate for the transmission between CAN-Master and CAN modules.

Com. Cycle Period, Sync. Window Length, Sync. COB-ID

PDOs (Process Data Object) are either for synchronous or asynchronous transmit modes. The **Com**munication **Cycle Period** [µsec] is the time interval in microseconds, in which the synchronization telegram with the unique number **Sync. COB-ID** (Communication Object Identifier) is transmitted.

This option must be activated if synchronization telegrams between CAN-Master and CAN-Slaves are to be sent.



The synchronous PDOs are transmitted directly after the synchronization telegram in the defined time slot **Sync. Window Length** [µsec]. If Com. Cycle Period and Sync. window length are 0, then no synchronization telegrams are transmitted.

If the synchronization telegram is defined in this dialog, timing jitters between the individual synchronization telegrams may occur due to the software architecture of the PLC runtime system.

Alternatively, the task synchronous transmitting of synchronization telegrams can be configured in the PLC program.

Node-Id

The **Node-Id** is the unique identification of the CAN module It corresponds to the number which is set between 1 and 127 on the CAN module itself. The Node-Id must be entered as a decimal number (not to be confused with the node number, which is used also in the PLC configuration)

Automatic startup

If the option **Automatic startup** is activated, all CAN modules will be automatically initialized and started when the PLC program starts up. If the option 'Automatic start' is not activated, the CAN modules must be manually started up in the PLC program.

Support DSP301, V4.01 and DSP306

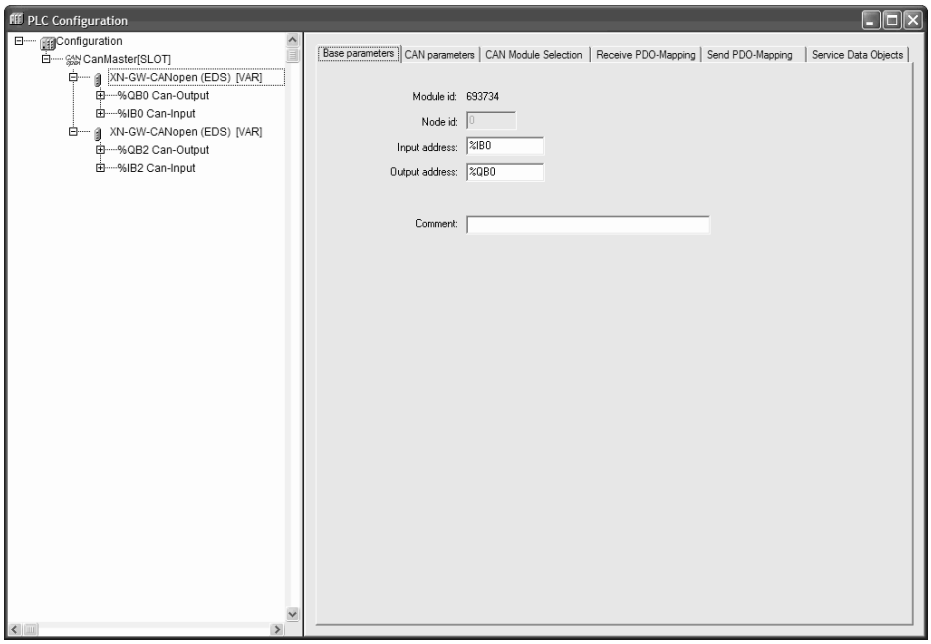
This option must be activated, if CAN modules are implemented in the PLC program, which support this CiA standard.

Heartbeat Master

Monitoring functionality: The CAN-Master transmits heartbeat telegrams with **Guard COB-ID** (Communication Object Identifier) at the appropriately defined interval. The default interval for the transmitting of heartbeat telegrams is 0 ms and is therefore deactivated.

5.3.2 Configuration of CAN-Nodes

5.3.2.1 Base parameters



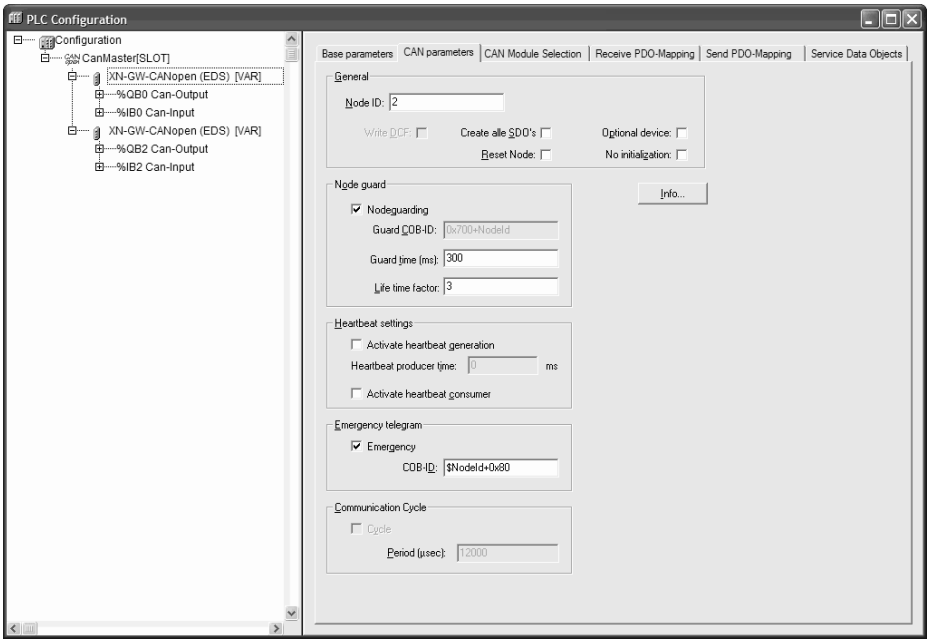
Node number

The node number is defined by an entry in the configuration file or - if there is no entry - by the position of the module in the configuration structure and is not editable (not to be confused with the node-Id, which can be defined by user).

Input address, Output address

This contains the IEC addresses starting from which the PDOs (Process Data Object) in the project can be addressed. It depends on the general settings and the definitions in the configuration file, which addresses are already predefined, which address mode is valid and whether the addresses can still be edited here.

5.3.2.2 CAN parameters



Node-Id

The **Node-ID** is the unique identification of the CAN module. It corresponds to the number which is set between 1 and 127 on the module itself. The Node-Id must be entered as a decimal number (not to be confused with the node number, which is used also in the PLC configuration).

Write DCF

If the option **write DCF** is activated, a DCF file will be created in the defined directory for the compiled files after an EDS file is inserted. The DCF file name is made up of the name of the EDS file and the corresponding Node-Id.

Create all SDOs

If the option **Create all SDOs** is activated, all SDOs will be created and transferred to the CAN-Node. Otherwise only SDOs are transferred which are different to the default value of the EDS file.

Reset node

If the option **Reset node** is activated, then the CAN-Slave will be reset before downloading the configuration by SDO command ‘restore all default parameters’ (Index 16#1011 Sub-Index 1 with value "Load", 16#23 11 10 01 6C 6F 61 64).

Optional device

If the option **Optional device** is activated, only in certain circumstances is the presence of the CAN node on the CAN bus checked after the start of the PLC program.

If the CAN node is not on the CAN bus, no node monitoring takes place. The node monitoring in the CAN diagnostic system takes place without the optional CAN node.

If the CAN node is connected on the CAN bus before the start of the PLC program, node monitoring is activated and evaluated in the CAN diagnostic system.

If the CAN node is connected on the CAN bus after the start of the PLC program, the CAN diagnostic system detects a node error.

No initialization

If the option **No initialization** is activated, the PLC program is starting without an initialization and starting of the CAN node.

Nodeguard settings

If the option **Nodeguarding** is activated, a guard telegram will be transmitted to the CAN module according to the interval set by **Guard Time** in milliseconds. If the CAN module does not then send a guard telegram with the given Guard **COB-ID** (Communication Object Identifier), it will receive the status "timeout".

As soon as the number of attempts (**Life Time Factor**) has been reached, the CAN module will receive the status "not OK". The status of the CAN module will be stored in the CAN diagnostic system and can be checked in the PLC program.

No monitoring of the CAN module will occur if the variables Guard Time and Life Time Factor are not defined (0).

→ Nodeguard functionality serves as alternative to heartbeat functionality.

Heartbeat settings

If the option **activate heartbeat generation** is activated, the CAN module transmits heartbeat telegrams with the given Guard **COB-ID** (Communication Object Identifier) according to the interval set by **Heartbeat Producer Time**. The CAN master expects this heartbeat telegram in this interval. If the CAN master does not receive this heartbeat telegram, the CAN module is detected as "not OK". The status of the CAN module will be stored in the CAN diagnostic system and could be checked in the PLC program.

If the option **activate heartbeat consumer** is activated, the CAN module expects heartbeat telegrams of the CAN master with the given Guard **COB-ID** (Communication Object Identifier) according to the interval set by **Heartbeat Master**.

→ Heartbeat functionality serves as alternative to nodeguard functionality

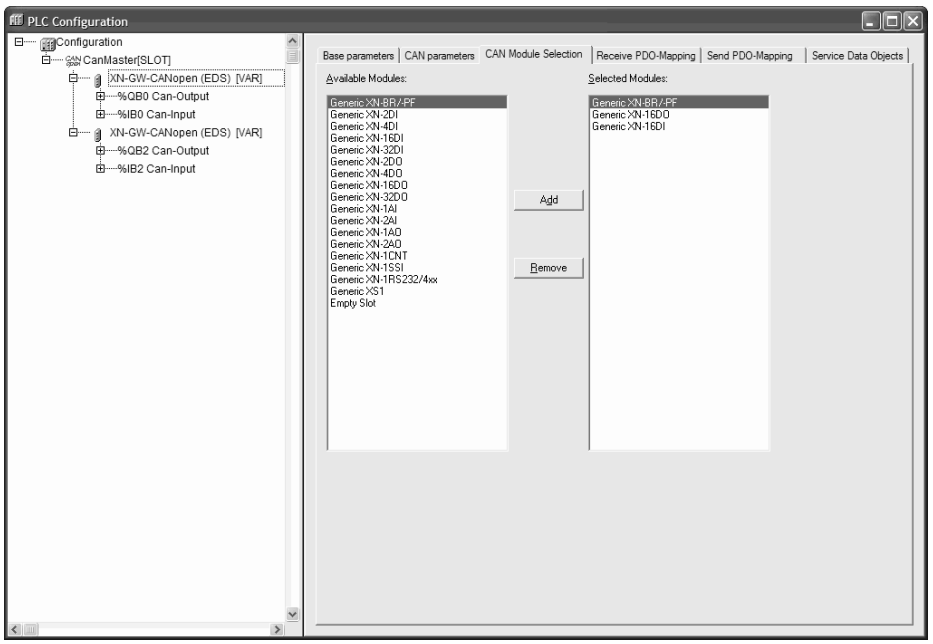
Emergency telegram

A module sends an emergency message, with a unique COB-ID, when there is an internal error. These messages, which vary from module to module, are stored in the diagnostic system and can be checked in the PLC program.

Info-Button

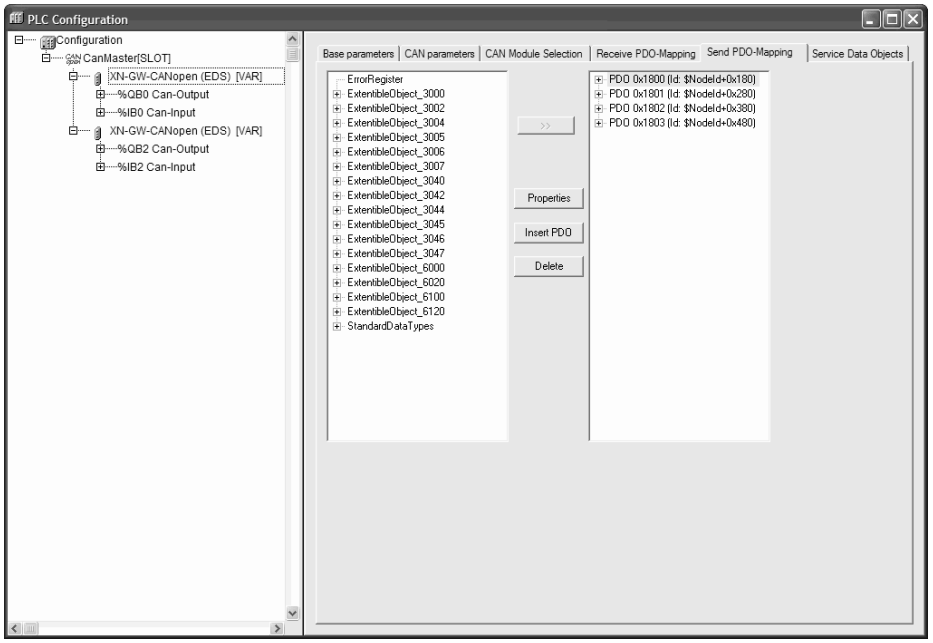
The entries "FileInfo" and "DeviceInfo" of the EDS or DCF file from the corresponding module manufacturer are hidden behind the Info button.

5.3.2.3 Module selection



If the inserted CAN module has a modular design and if it supports the appropriate standards (**DSP 301, V4.01 and DSP 306**), then the tab **CAN Module selection** appears. The configuration of the appropriate structure can be made by adding with **Add** button or removing with **Remove** button.

5.3.2.4 PDO-Mapping (Receive, Send)



The tabs **Receive PDO mapping** and **Send PDO mapping** in the configuration dialog allow the "mapping" of the module described in the EDS file to be changed.

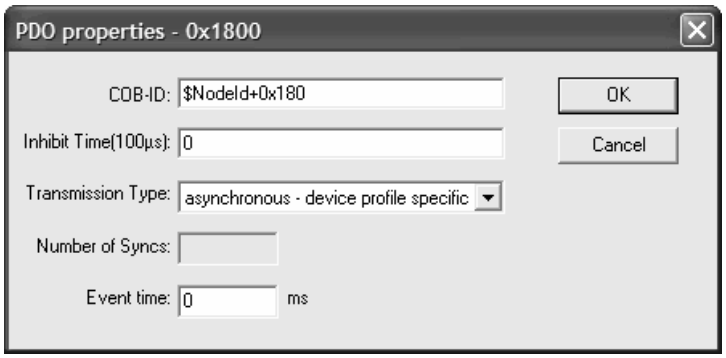
All "mappable" objects in the EDS file are located on the left side and can be added to the right side to the PDOs (Process Data Object) with **>>** button or removed again with **Remove** button.

The StandardDataTypes can be inserted to create empty spaces in the PDO.

The **Insert Element** button can be used to create further PDOs and to add appropriate objects to them. The allocation of inputs or outputs to the IEC addresses can be achieved via the inserted PDOs. The setting which has been made in the PLC configuration will become visible after the dialog is left. The individual objects can be defined there with symbolic names.

→ The communication channels send and receive are from the point of view of the CAN module. This means that inputs configured in the PLC program are from the point of view of the CAN module in the send channel and outputs are in the receive channel

The standard set properties of the PDOs can be edited with **Properties** button. The field appears in grey and cannot be edited if an option is not supported by the module or if the value cannot be changed.



COB-ID

Each PDO message requires a unique **COB-ID** (Communication Object Identifier).

Inhibit Time

The Inhibit Time is the minimum time between two messages from this PDO. This is to prevent PDOs which are sent when the value is changed from being sent too often.

Transmission Type, Number of Sync's, Event-Time

Transmission Type offers you a selection of possible transmission modes for these PDOs:

acyclic – synchronous : the PDO will be transmitted synchronously but not periodically.

cyclic – synchronous : the PDO will be transmitted synchronously, whereby the **Number of Sync's** specifies the number of synchronization messages between two transmissions of this PDO.

cyclic – RTR only : the PDO will be updated after each synchronization message but not sent. It is only sent when there is an explicit request to do so (Remote Transmission Request).

asynchronous – RTR only : the PDO will only be updated and transmitted when there is an explicit request to do so (Remote Transmission Request).

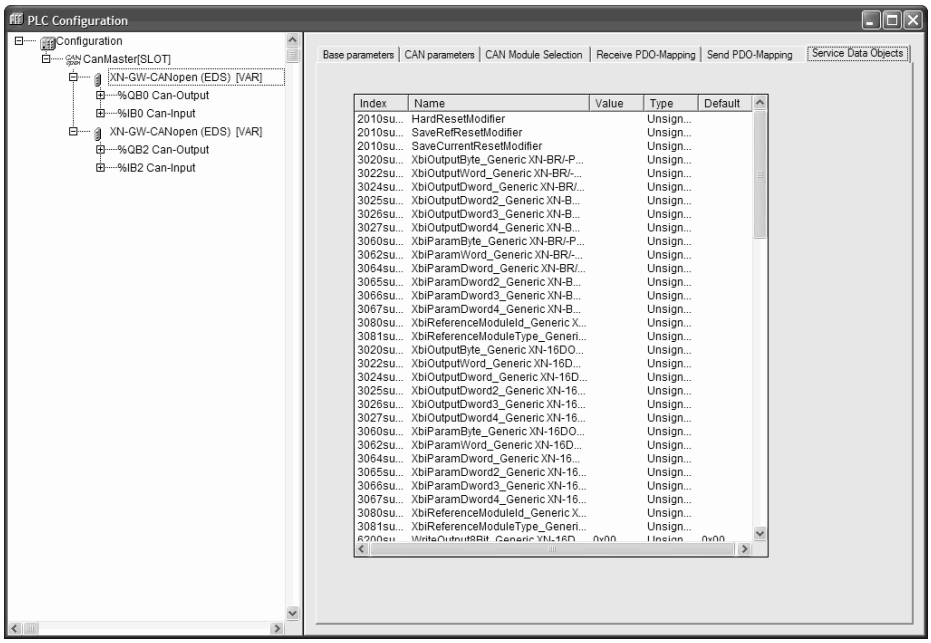
asynchronous – manufacturer specific and **asynchronous – device profile specific** : the PDO will only be transmitted when specific events occur. An addition event can be defined with the **Event-Time**. Enter here in milliseconds (ms) the interval between two transmissions.

5.3.2.5 Service Data Objects

The following is a list of all objects in the EDS or DCF file which are in the area of the Index 0x2000 to 0x9FFF and which are marked as writable.

The properties **Index, Name, Value, Type** and **Default** are displayed for every object. The value can be changed. Mark the value and press the <Space bar>. After making the change confirm the new value with <Enter> or reject it with the <Escape> pushbutton.

The set values are transmitted in the form of SDOs (Service Data Object) to the CAN modules at the initialization of the CAN bus



5.3.3 Bus diagnostic

Various function libraries are available for configuring the bus diagnostics in the PLC program.

→ Please refer to the detailed function descriptions in the relevant documentation of the function libraries.

5.4 Configuration as CAN-Device

A device which is programmed with the PLC programming tool "XSoft CoDeSys-2" can appear and be used in a CAN network as CANopen Slave (called in the following CAN-Device).

The parameter manager, PLC configuration and library functions make the following options available:

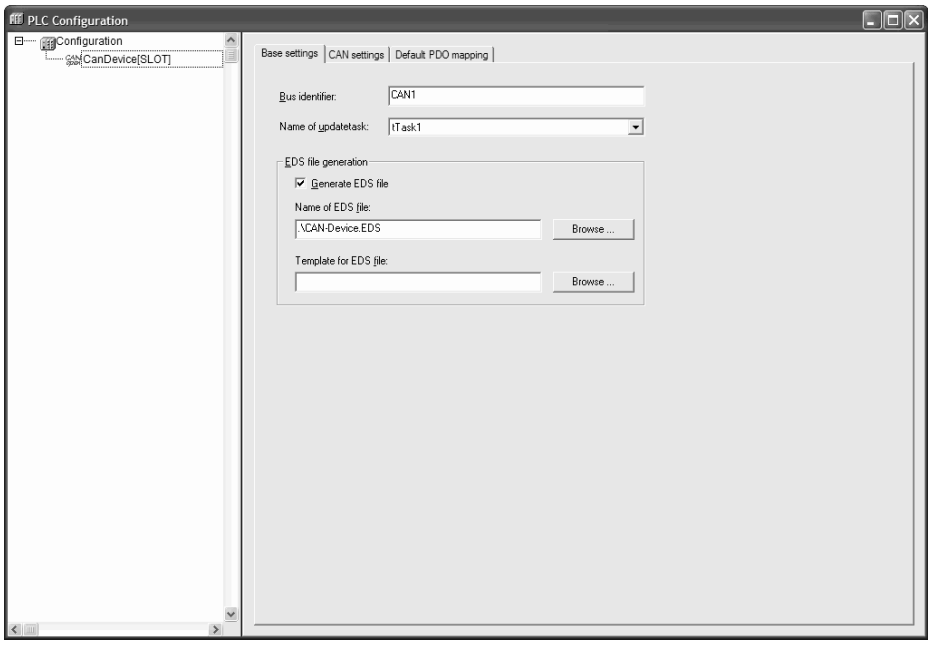
- Configuration of the variable or parameter lists, which are used for data exchange between CAN-Device and CAN-Masters (parameter manager, provide the object listing).
- Configuration of the nodeguard/heartbeat functionality, the emergency message, the node number Node-Id and the baud rate.
- Configuration of default PDO mapping based on the variable or parameter lists.
- Generation of the EDS file, which describes the CAN-Device and which can be inserted in the PLC program of the CAN-Master.
- Library functions for CAN-Device to monitor and administrate the object listing

The following functions are not available:

- dynamic SDO or SDO identifier
- SDO block transfer
- Implicit generation of emergency messages. Emergency messages must always be generated by the application. The library provides for this an FB which can be used by the application.
- Dynamic changes of the PDO properties/runtime properties

5.4.1 Configuration of CAN-Device

5.4.1.1 Default setting



Bus identifier

Currently not used!

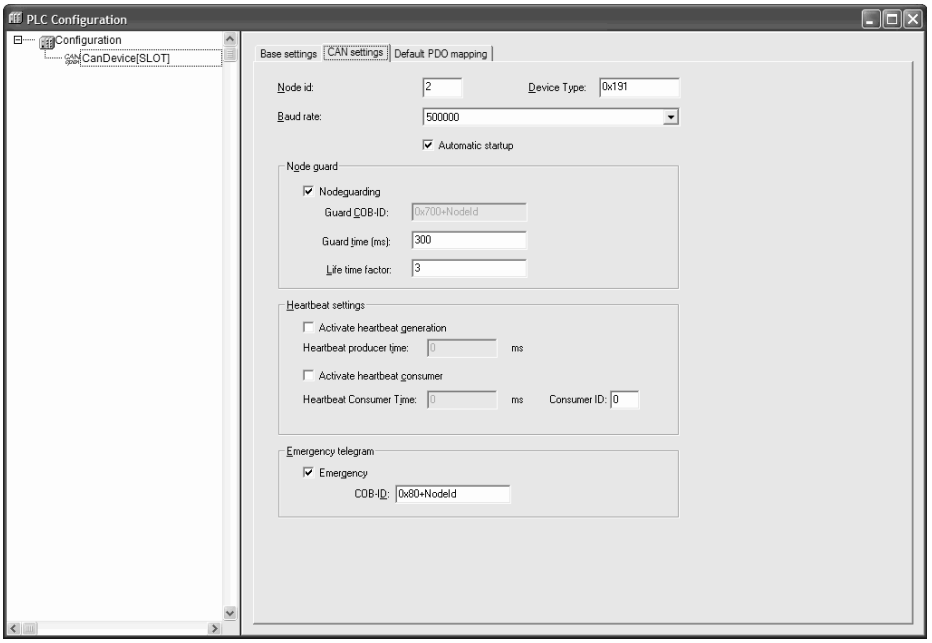
Name of update task

Name of the task, in which the CAN-Device is called.

EDS file generation

If the option **EDS file generation** is activated, it will generate an EDS file with the **Name of the EDS file** in order to be able to use the current configuration later in any master configuration.

5.4.1.2 CAN settings



Node-Id

The **Node-ID** is the unique identification of the CAN-Device. It corresponds to the number which is set between 1 and 127 on the CAN-Device itself. The Node-Id must be entered as a decimal number (not to be confused with the node number, which is used also in the PLC configuration).

→ The Node-ID can be overlaid by means of library functions in the PLC program. Thus it is possible to write a PLC program for several CAN-Devices, without changing the PLC configuration of the CAN-Devices.

Device type

The **Device type** (default value of object 0x1000) of the device is predefined with 0x191 (standard IO Device) and can be changed by the user.

Baud rate

Baud rate for the transmission between CAN-Master and CAN-Device

Automatic startup

If the option **Automatic startup** is activated, the CAN bus will be automatically initialized and started by PLC program start.

Nodeguard settings

Parameterization of **Guard COB-ID** (Communication Object Identifier), **Guard Time** in milliseconds and **Life Time Factor**. These parameters are inserted as default values into the EDS file of the CAN-Device and can be changed afterwards in the PLC configuration of the CAN-Master.

→ Nodeguard functionality serves as an alternative to heartbeat functionality.

Heartbeat settings

Parameterization of **Heartbeat Producer Time** and **Heartbeat Consumer Time** in milliseconds. These parameters are inserted as default values into the EDS file of the CAN-Device and can be changed afterwards in the PLC configuration of the CAN-Master.

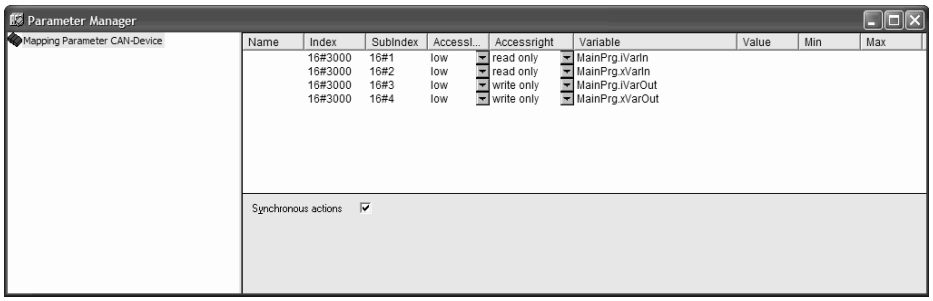
→ Heartbeat functionality serves as an alternative to nodeguard functionality.

Emergency telegram

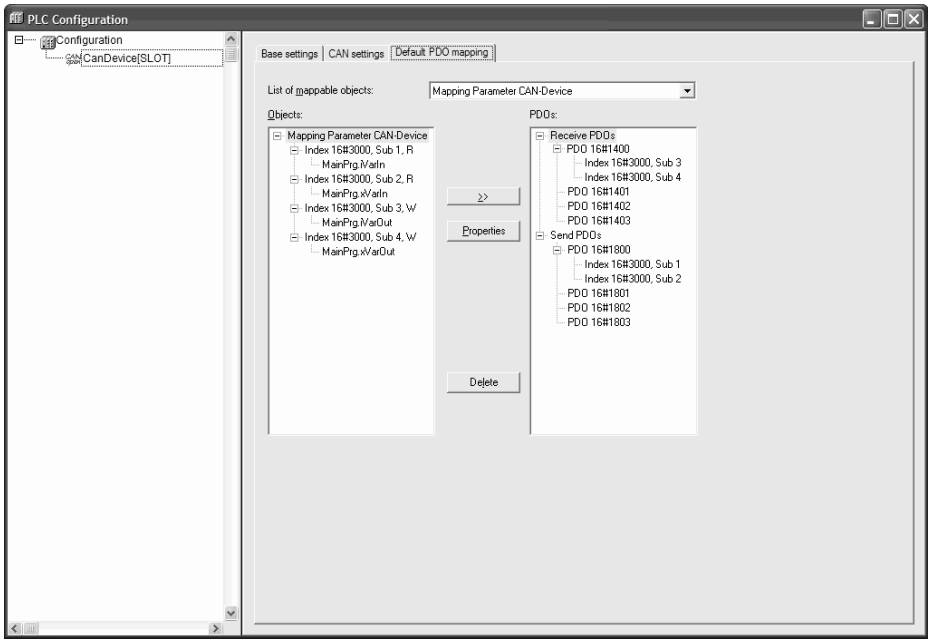
Parameterization of **Emergency** message with a unique **COB-ID**. These parameters are inserted as default values into the EDS file of the CAN-Device and can be changed afterwards in the PLC configuration of the CAN-Master.

Default PDO mapping

The variable or parameter lists are created in the Parameter Manager, which are available afterwards in the PDO mapping of the CAN-Device.



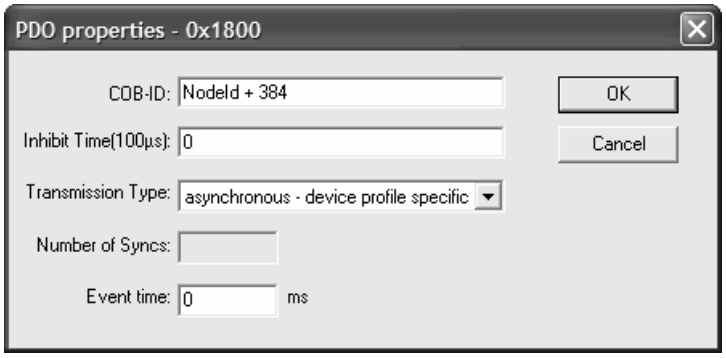
→ In order to be able to open the Parameter Manager, it must be activated and defined with correct index/subindex ranges in the map 'Network functionality' of the target settings.



The **Default-PDO-Mapping** tab allows the entries which are defined in the Parameter Manager to be assigned/modified to the "Mapping" of the CAN-Device.

All "mappable" objects are located on the left side and can be added in the right side to the PDOs (Process Data Object) with button **>>** or removed again with button **Remove**.

The properties of the PDOs can be edited with button **Properties**. These can be changed afterwards in the PLC configuration of the CAN-Master.



5.4.2 Configuration of CAN-Device in CAN-Master

After insert of CAN-Device, PLC configuration makes the following options available:

- Configuration of the nodeguard/heartbeat functionality, the emergency message and the node number Node-Id.
- Configuration of PDO mapping based on the default PDO mapping if the EDS file.

Please refer to the detailed information in the chapter 'Configuration of CAN-Nodes' (➔ Chap. 5.3.2).

5.5 Configuration as Profibus-Master

The PLC programming tool supports a hardware configuration according to Profibus-DP Standard. This requires a configuration file which allows Profibus modules to be inserted.

All GSD files which are stored in the defined configuration files directory, can be integrated, edited and displayed in the PLC configuration. The configuration options of a Profibus module are described in the GSD file.

The modules are assigned a configuration, which describes the timing and error behaviour of the transmission.

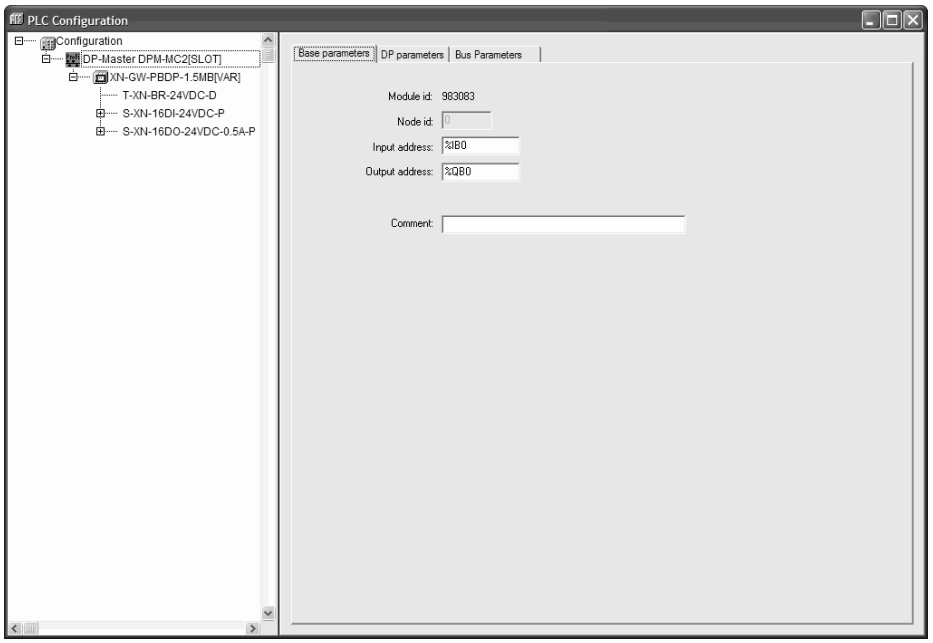
Objects defined in the PLC configuration for which the inputs or outputs are not used in the PLC program are not updated by default in the process image.

5.5.1 Configuration of Profibus-Master

The following Profibus-Masters is implemented in the PLC configuration:

Designation	Description
DP-Master CMM-MPI (MPDP4D09.GSD)	Profibus-Master DP-V1 for Profibus DP / FMS and MPI

5.5.1.1 Base parameters



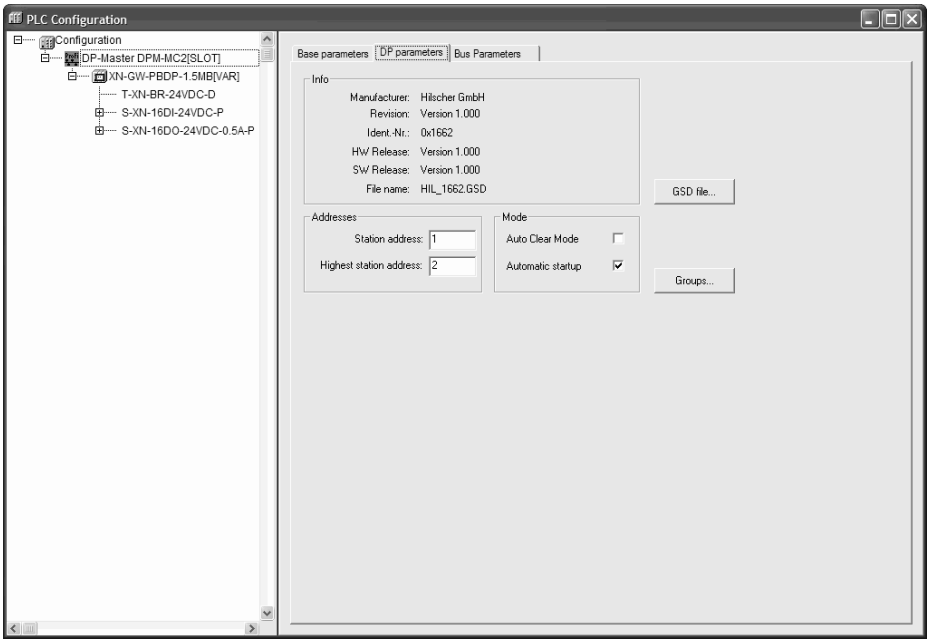
Node id

The node identification is defined by an entry in the configuration file or - if there is no entry - by the position of the module in the configuration structure and is not editable (not to be confused with the node ID, which can be defined by user).

Input address, Output address

This contains the IEC addresses starting from which the inputs or outputs in the project can be addressed. It depends on the general settings and the definitions in the configuration file, which addresses are already predefined, which address mode is valid and whether the addresses can still be edited here.

5.5.1.2 DP parameters



Station address

The **Station address** is the unique identification of the Profibus-Master. It is predefined with the value 1 and does not have to be edited by the user.

Highest station address

It corresponds to the highest station address existing on the Profibus (adjustable by user, or calculated by PLC programming tool).

→ Make sure that a correct HSA is entered!
Following HSAs are supported: 15, 31, 63 and 126

Auto Clear Modus

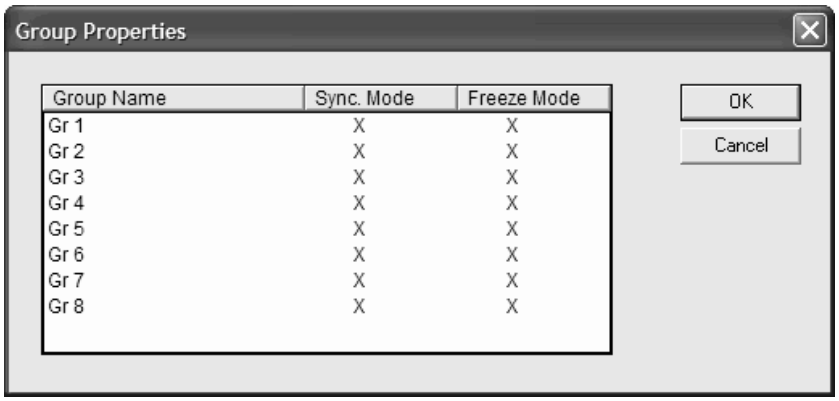
Each Profibus-Slave has a watchdog, which monitors whether the Profibus-Slave is addressed cyclically by the Profibus-Master. If this is not the case, the Profibus-Slave switches autonomously into the safe operating state. If this means a high safety risk, the option **Auto Clear Mode** causes the Profibus-Master to switch automatically from operating state "operate" to operating state "clear". This in turn causes all other Profibus-Slave to be switched to the operating state "clear". Digital outputs are set to 0 and analogue outputs are set to the configured substitute values (FailSafe).

Automatic startup

If the option **Automatic startup** is activated, the Profibus will be automatically initialized and started.

→ Must be always selected (a manual start is not supported).

Groups



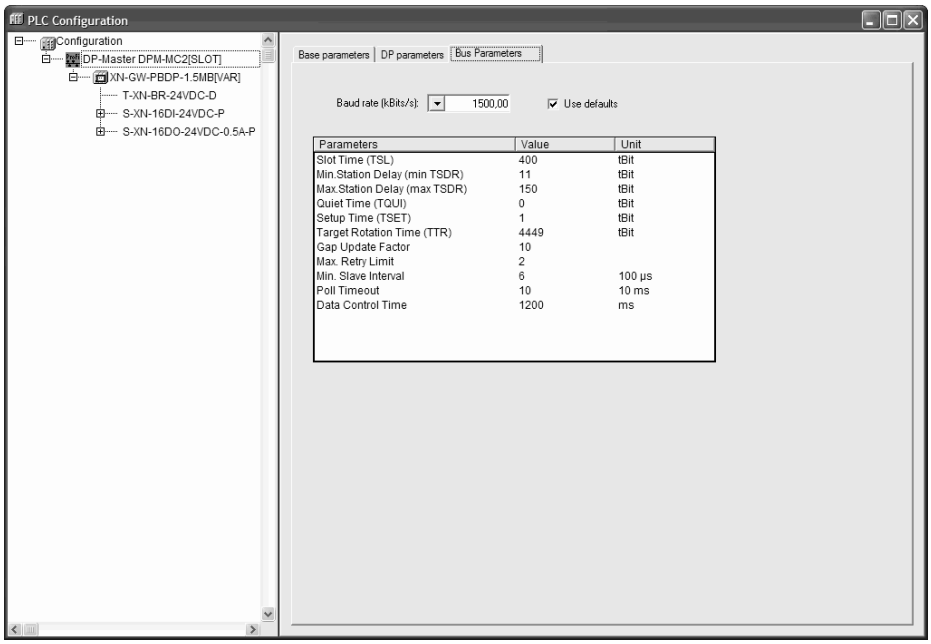
→ Freeze- and Sync-Mode are not supported.

The **Groups** button opens the 'Group properties' dialog. The Group properties pertain to the slaves assigned to the master. Up to eight groups can be set up. For each group, enter whether it is to operate in **Freeze** mode and/or **Sync** mode. By assigning the slaves to various groups, data exchange from the master can be synchronized via a global control command. With a Freeze command, a master instructs a slave or a group to "freeze" inputs in their instantaneous state and to transfer this data in the next data exchange. With a Sync command, the slaves are instructed to synchronously switch to the outputs at the next Synch command all data received from the master following the first command.

In the dialog 'Group properties' can be edited the group names and activated and/or deactivated the appropriate transmit modes.

5.5.1.3 Bus parameters

The bus parameters describe the timing of the communication.



Baud rate

Baud rate for the transmission between Profibus-Master and Profibus-Slaves

Use setpoint values

If **Use defaults** is activated, a default parameter set is used for selected baud rate. If this option is not activated, the parameters can be changed manually.

→ This option does not take the configured I/Os into account. If optimized parameters for the bus topology are needed, then these parameters must be determined with an external Profibus configuration tool (e.g. Sycon supplied by Hilscher).

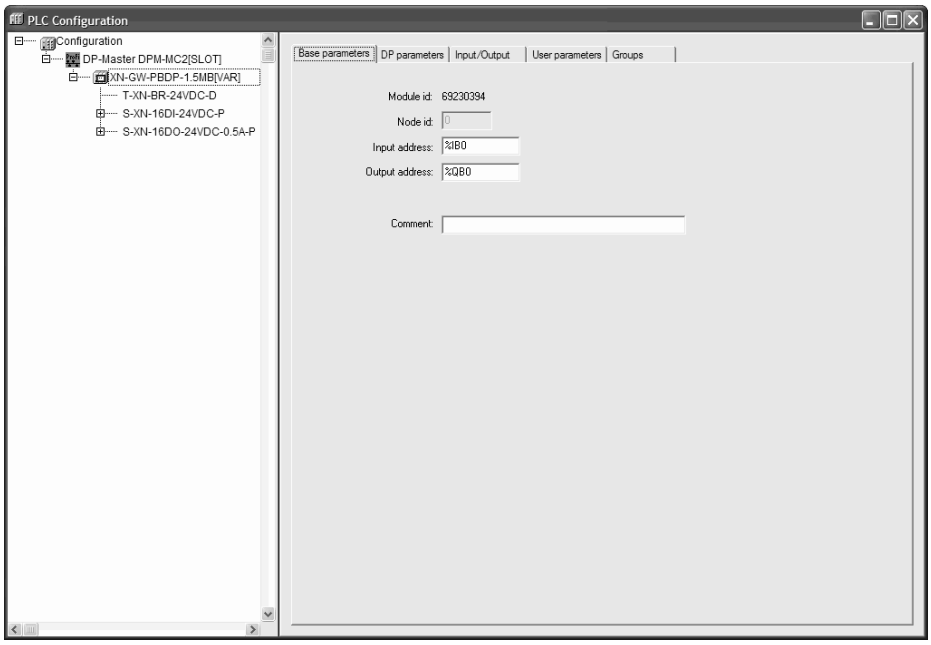
Parameters

Name, **value** and **unit** are indicated for each object. The value can be changed. Mark the value and press <Space>. After any changes press <Return> to confirm or press <Escape> to reject the value.

Parameter	Description
Slot Time (TSL)	Maximum time during which the master waits, after sending a request message, for the receipt of the first character of the slave's reply message.
Min. Station Delay (Min. TSDR)	Minimum response time, after which a Slave may reply.
Max. Station Delay (Max. TSDR)	Maximum response time, within which a Slave must reply
Quiet Time (TQUI)	Idle period which must be taken into account during conversion of NRZ (Non Return to Zero) signals to other codings (changeover time for repeater).
Setup Time (TSET)	Time, which may lapse between the receipt of a telegram and the following reaction on the Slave.
Target Rotation Time (TTR)	Token cycle time setting: Projected time interval in which a master should receive the token. Result of the sum of the token stop times of all masters on the bus.
Gap Update Factor (GAP)	GAP update factor G: Number of bus cycles after which the master's GAP (address range from its own bus address to the address of the next active station) searches for an additional, newly inserted active station.
Max. Retry Limit	Maximum number of repeated request attempts by the master when it has not received a valid response from the slave
Min. Slave Interval	Time between two bus cycles in which the slave can process a request from the master (time basis 100µs). The value entered here must be checked against the respective specifications in the slave's GSD file.
Poll Timeout	Maximum time after which the master's reply in a master-master communication must be retrieved by the requester (Class 2 DP master) (time basis 1ms).
Data Control Time	Time in which the master reports its status to the slaves assigned to it. At the same time, the master monitors whether at least one data exchange each has taken place with the slaves within this period, and updates the Data_Transfer_List.

5.5.2 Configuration of Profibus-Slaves

5.5.2.1 Base parameters



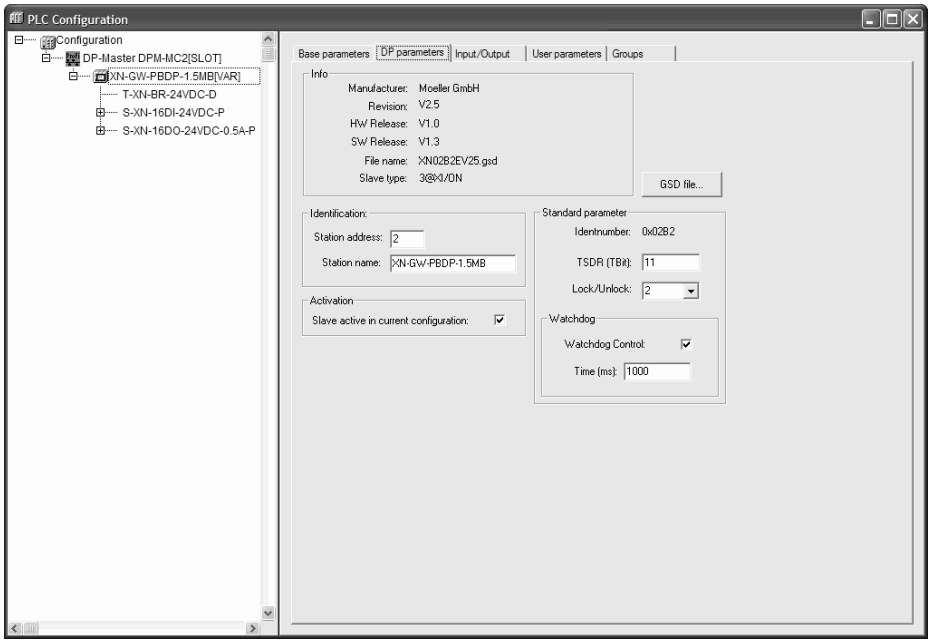
Node id

The node identification is defined by an entry in the configuration file or - if there is no entry - by the position of the module in the configuration structure and is not editable (not to be confused with the station address, which can be defined by user).

Input address, Output address

This contains the IEC addresses, starting from which the inputs or outputs in the project can be addressed. These addresses refer to the module. It depends on the general settings and the definitions in the configuration file, which addresses are already predefined, which address mode is valid and whether the addresses can still be edited here.

5.5.2.2 DP parameters



Station address

The **Station address** is the unique identification of the Profibus module. It corresponds to the number which is set between 0 and 126 on the Profibus module itself. The station address must be entered as a decimal number.

Activation

If the option **Activation** is deactivated, the configuration data is transferred to the Profibus-Slave during Download, but a data exchange does not occur.

Standard parameter

The **Identnumber** is assigned by the PNO and is the unique identification number for this device type. It allows unambiguous reference between Profibus-Slave and the corresponding GSD file. The parameter **TDSR** (**T**ime **S**tation **D**elay **R**esponder) corresponds to the response time after which the Profibus-Slave may respond at the earliest to the Profibus-Master. Time unit for the transmission of a bit via Profibus: Reciprocal value of the data baud rate: e.g. 1 TBit at 12MBaud=1/12.000.000 Bit/sec=83ns. The parameter **Lock/Unlock** enables the Profibus-Slave to be locked or released for other Profibus-Masters:

0: min.TSDR and slave-specific parameters may be overwritten

1: Slave released to other masters

2: Slave locked to other masters, all parameters are accepted

3: Slave released to other masters

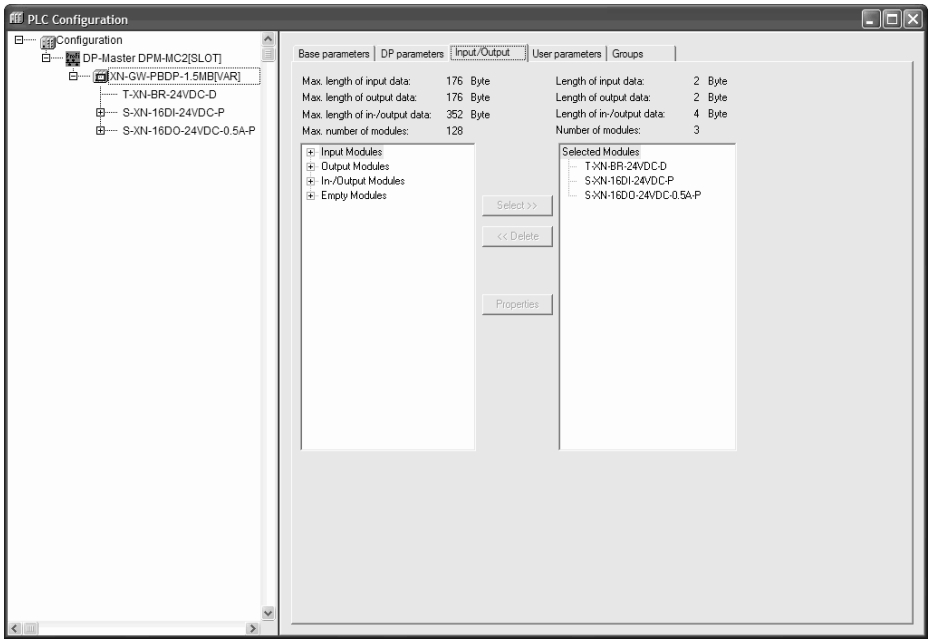
Watchdog

If the option **watchdog control** is activated, the defined watchdog time applies. If the Profibus-Slave has not been accessed by the Profibus-Master within this time, it is reset to its initialization state.

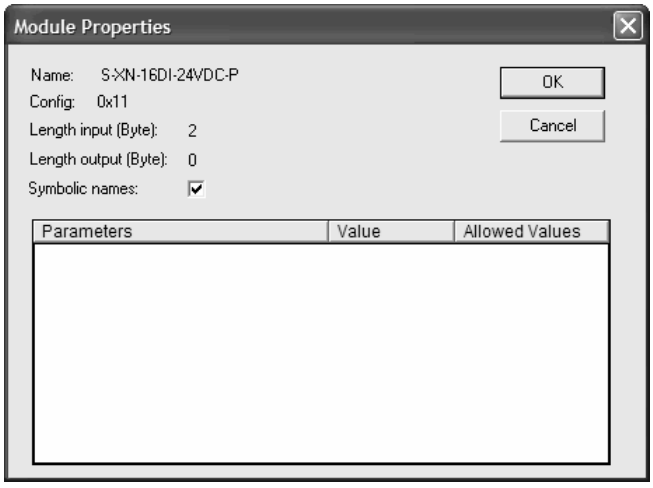
→ The same watchdog control time must be set on all slaves.

5.5.2.3 Module selection

If the inserted Profibus module has a modular design and if it supports the appropriate standards, then the **Input/Output** tab appears. The configuration of the appropriate structure can be made by adding with the **Add** button or removing with the **Remove** button.



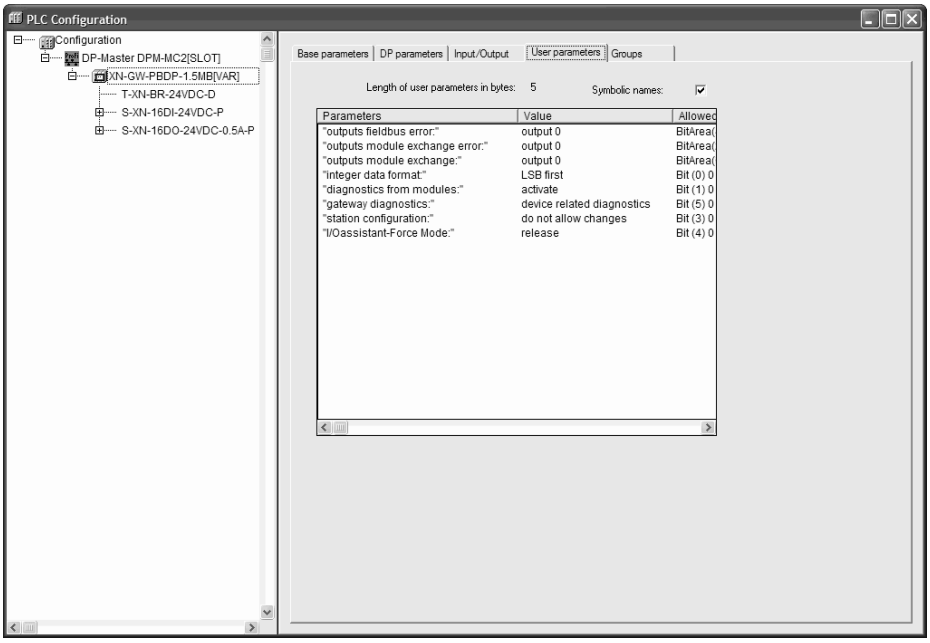
The **Properties** button opens the 'Module properties' dialog for the input or output module currently selected in the left or the right list. It shows the name, the configuration (module description coding according to PROFIBUS standard) and the input and output lengths of the module in bytes. If the module description in the GSD file contains specific parameters in addition to the standard set, these are listed here with their values and range of values. If the **Symbolic names** option is activated, the symbolic names are then used.



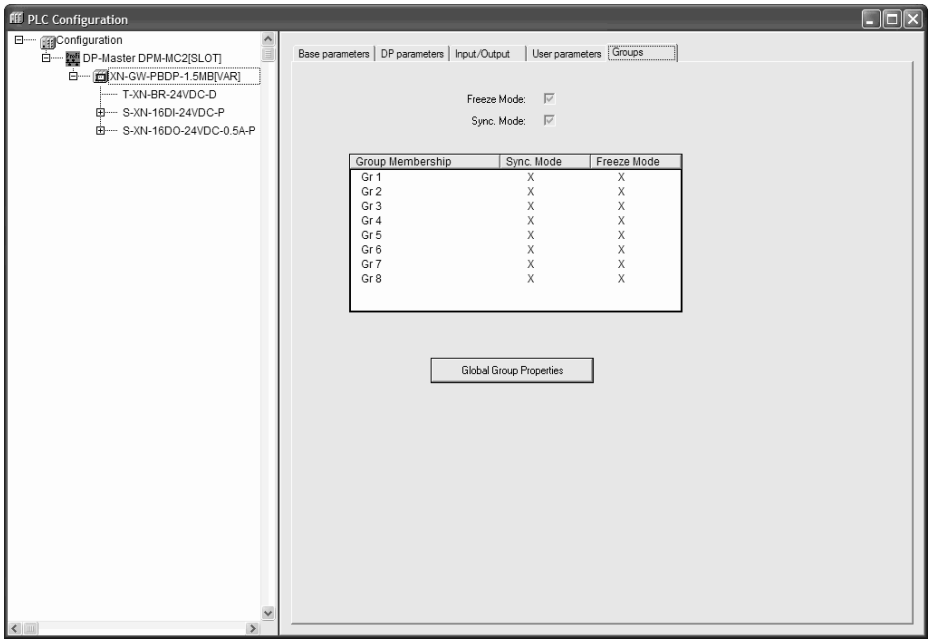
5.5.2.4 User parameters

This tab lists various extended parameters of a Profibus-Slave defined in the GSD file. The column **Parameters** shows the name of the parameter. The parameter values entered in column **Value** can be changed. The **Value range** is also specified.

If symbolic names are also specified for the parameters in the GSD file, the option **Symbolic names** can be activated, so that the values can be displayed with these names. The Length of user parameters is also given above the table as information.



5.5.2.5 Groups



→ Freeze- and Sync-Mode are not supported.

This dialog is used for assigning the Profibus-Slave to one or more of the eight possible groups. The universally applicable group properties (Sync. Mode and/or Freeze Mode), on the other hand, are defined during configuration of the Profibus-Master's properties. This dialog can also be reached via the **Global Group Properties** button.

The group(s) to which the Profibus-Slave is assigned are marked with a plus sign. The assignment to or removal from a group is accomplished by selecting the group name in the Group Membership column and pressing 'Add slave to group' or 'Remove slave from group' with the right mouse button, or by clicking again with the mouse to the left of the group name.

A Profibus-Slave device can only be assigned to those groups whose properties it supports. The relevant properties of each slave (Sync. Mode / Freeze Mode) are displayed above the table. The modes supported by the device are checked.

5.5.3 Bus diagnostic

For configuring the bus diagnostics in the PLC program various function libraries are available.

Please refer to the detailed function descriptions in the relevant documentation of the function libraries.

5.6 SmartWire-DT configuration

The PLC programming environment supports the configuration and operation of a SmartWire-DT network.

The adding of SmartWire-DT slaves requires the use of an 'SWD-Master.gsd' configuration file that contains the relevant data for the slaves. This file is located in the folder 'SWDConf'.

→ A general description of the SmartWire-DT system is provided in the document "SmartWire-DT System", MN05006002Z-EN.

→ Function descriptions and information about the input/output data volume of the individual SmartWire-DT slaves is provided in the document "SmartWire-DT Units", MN05006001Z-EN.

System requirements

PLC target system	PLC runtime system	Operating system (OS)
≥ XC-152-V2.3.9 SP3	≥ PLCWinCE V 2.4.10 (xxx)	Windows CE 5.0 ≥ Image Release 2.26.0 (xxx)

→ The SWD project configuration is only valid if at least one variable is used (referenced) in the PLC program.

→ Objects defined in the PLC configuration for which the inputs or outputs are not used in the PLC program are not updated by default in the process image.

→ The update time of SmartWire-DT IOs depends on the total number of SmartWire-DT IOs and the baud rate used.

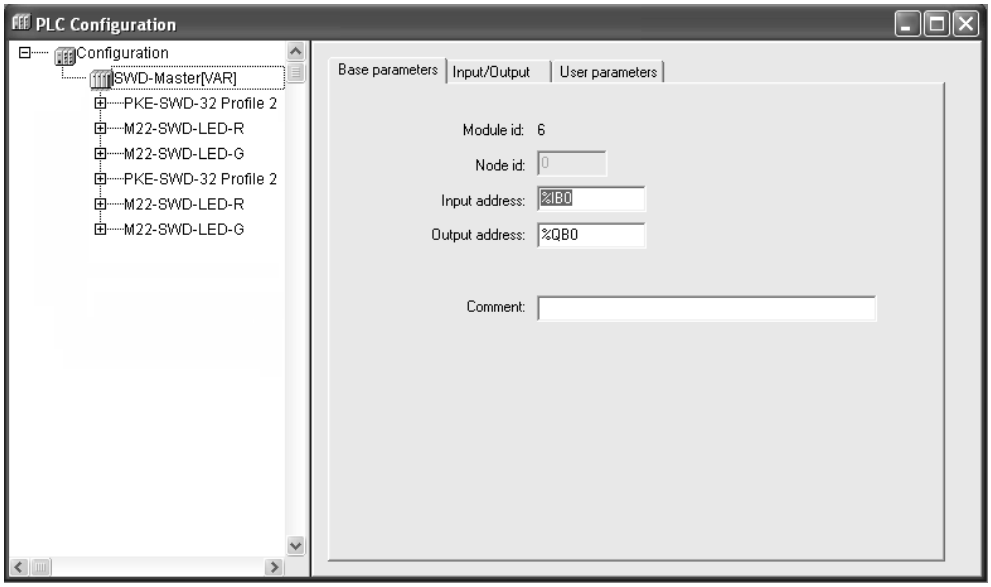
5.6.1 Configuration of the SWD master and SWD module selection

The following SWD master is provided in the PLC configuration:

Designation	Description
SWD Master	SWD coordinator

In the PLC configuration choose Append subelement → SWD master….

5.6.1.1 Base parameters



Node id

The node identification is defined by an entry in the configuration file or - if there is no entry - by the position of the module in the configuration structure and is not editable.

Input address, Output address

Here you enter the IEC addresses, starting from which the inputs or outputs in the project can be addressed. The general settings and the definitions in the configuration file determine which addresses are already predefined, which address mode is used in the configuration, and whether the addresses can still be edited here.

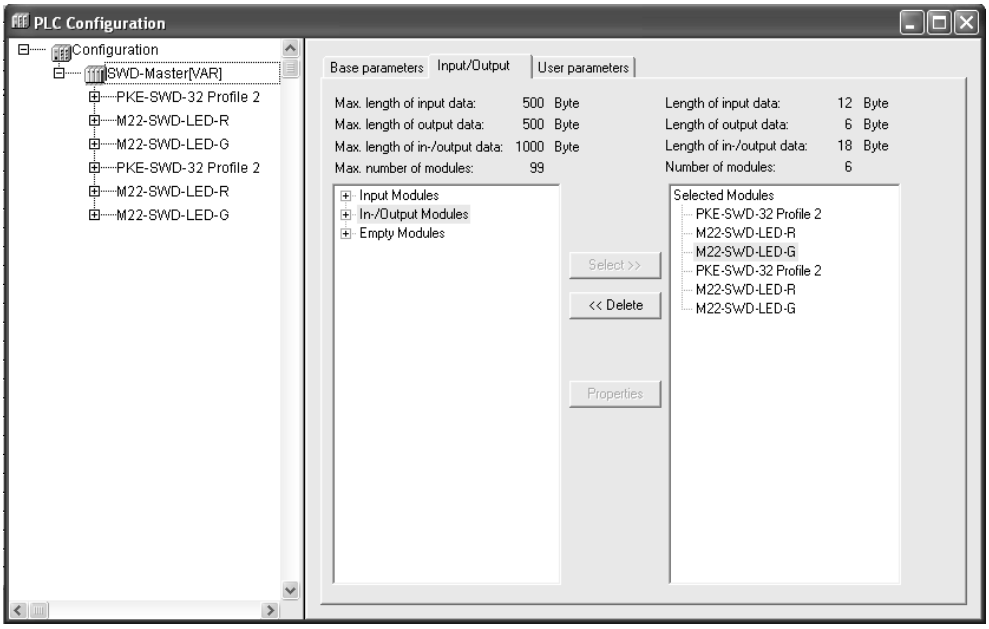
5.6.1.2 Slave selection

Change over to the **Inputs/Outputs** tab.

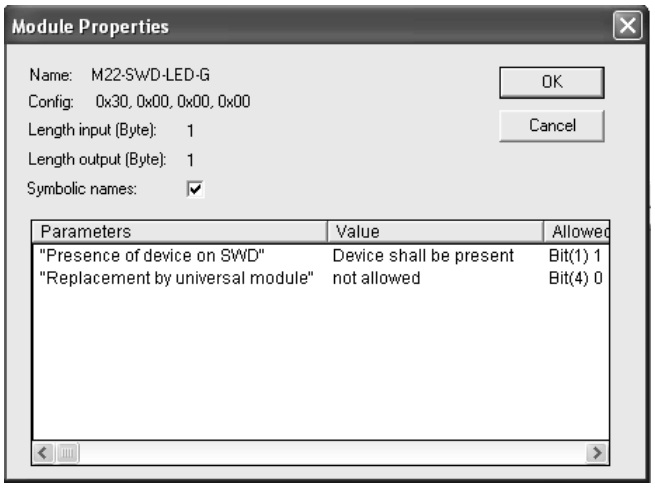
The slaves (modules) can then be added to the inserted SmartWire-DT master. The known slaves appear in the left-hand window of the Inputs/Outputs tab**.** The configuration of the appropriate structure can be made by **selecting** or **removing** SmartWire-DT slaves.

Please heed of the correct sequence. The modules must be configured just as they will be arranged afterwards in your application.

A total of 99 SmartWire-DT slaves can be connected on a SmartWire-DT line. The length of the SmartWire-DT network can be extended up to 600 meters.



Clicking the **Properties** button opens the Module Properties dialog of the currently selected SmartWire-DT slave. It shows the name, the Config (coding of the IO image in accordance with the PROFIBUS standard) and the input and output length of the module in bytes.



The parameters 'Presence of device on SWD' and 'Replacement by universal module' are standard parameters for all SmartWire-DT slaves.

The 'Presence of device on SWD' entry determines whether a slave in the SmartWire-DT network is classed as necessary (Device shall be present) or as optional (Device may not be present). The absence of a necessary slave switches the SmartWire-DT network to the FAILSAFE state. The absence of an optional slave has no effect on the state of the SmartWire-DT network. The default setting is 'Device shall be present'.

The entry 'Replacement by universal module' defines whether a slave in the SmartWire-DT network can be replaced by a universal slave. The default setting is 'not allowed'.

If the slave contains additional parameters, these are also listed here with the corresponding value and value range. If the **Symbolic names** option is activated, the possible settings are represented with symbolic names.

5.6.1.3 User parameters

Move to the **User parameters** tab

This is where you carry out the general settings of the SmartWire-DT network.

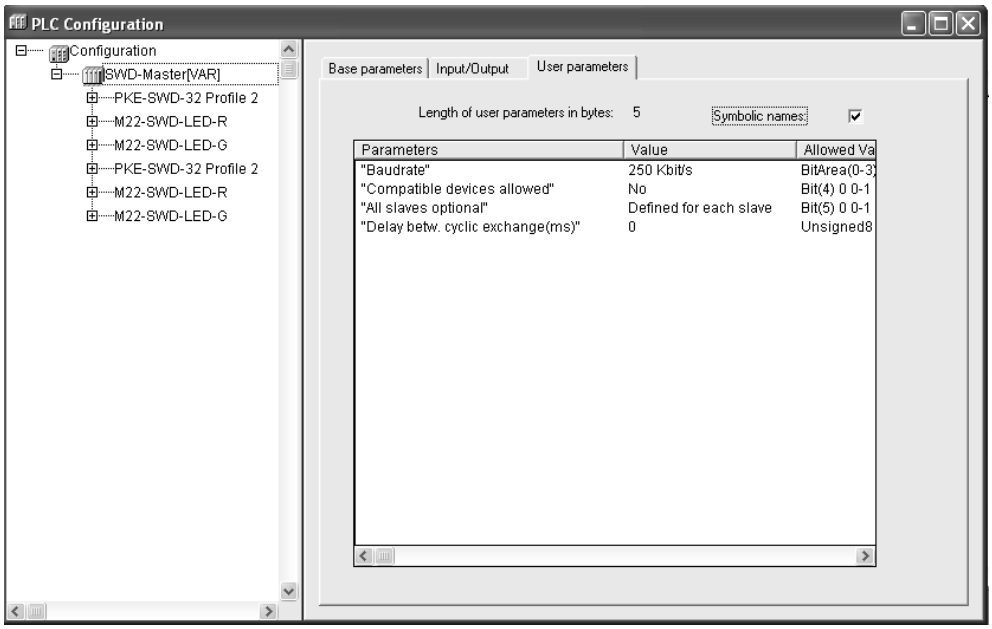
The **'Baudrate'** parameter is used to set the SmartWire-DT baud rate. Permissible values: 125kBaud / 250kBaud (default setting).

The parameter **'Compatible devices allowed'** defines whether a device that is compatible with the configured device is also allowed instead of it.
Default setting = No

For example, an M22-SWD-K22LED-W is used in the network instead of a configured M22-SWD-K11LED-W. → Only slaves with an identical "Family code" and a higher "Device Code" are accepted as compatible slaves.

The parameter **'All slaves optional'** enables all configured SmartWire-DT slaves to be classed as optional slaves via a switch.
Default setting = Defined for each slave

'Delay betw. Cyclic exchange(ms)' is used to set a delay time between two cyclical data telegrams.
Default value = 0



5.6.2 Bus diagnostic

Each SmartWire-DT slave always provides status information in its first input byte:

Bit 0-3: Slave-specific user data.

Bit 4: TRUE = Extended diagnostics message present for the slave.

Bit 5: Reserved.

Bit 6: TRUE = The slave is present.

Bit 7: TRUE = The slave is replaced by a universal module.

Other status messages on the SmartWire-DT coordinator and the connected slaves can be called using the functions and function blocks of the SWD_Util library. Description in document MN04802103Z-EN.

5.6.3 Acyclic Communication

Acyclic communication with SmartWire-DT slaves can be started with the function block SWD_ACYCLICREQUEST in the SWD_Util library.

5.6.4 SmartWire-DT LEDs

Three LEDs are provided on the XC150-SWD hardware. These are called POW, CONFIG and SWD.

POW-LED

State	Description
OFF	Power supply of the local SmartWire-DT not present.
Green continuous light	Power supply of the local SmartWire-DT ASIC OK.

CONFIG-LED

For a data exchange the created configuration (= designed configuration) must match the configuration actually present in the SWD master (target configuration). The result of the check is indicated via the SmartWire-DT Config LED.

State	Description
OFF	No project configuration or faulty target configuration (see SWD LED).
Green continuous light	Project configuration identical to stored target configuration.
Flashing (green)	Project configuration compatible with stored target configuration.
Red continuous light	Project configuration and stored target configuration are not compatible.

SWD-LED

If a configuration is stored in the SWD master, each time the supply voltage is switched, a check is made whether the slaves actually found on the network match the stored gateway target configuration. The result of the check is displayed via the SWD LED.

State	Description
OFF	No target configuration present.
flashing orange	Reading and storing the actual slave list as a new target configuration.
Green continuous light	The slaves found on the SmartWire-DT line match the target configuration.
Flashing (green)	Slave search / slave addressing is active.
Red continuous light	Short-circuit on the 15 V power supply or no SmartWire-DT slave found.
Flashing (red)	The slaves found on the SmartWire-DT line do not match the target configuration, or a SmartWire-DT slave configured as necessary is missing.

5.6.5 Config button

The current configuration can be stored as the target configuration via the Config button on the XC150-SWD hardware. This function is provided as long as the SmartWire-DT network is not in the states "SWD_RUN_FAILSAFE" or "SWD_RUN_NORMAL" (Config LED not green).

The target configuration must be carried out every time a slave is exchanged on the SmartWire-DT network.

The target configuration can also be read and stored via the browser command "swd_setnew".

5.6.6 Browser commands

The XC150 SmartWire-DT master supports the following browser commands:

Command	Description
?	Gets the list of implemented and available commands
swd_targetconfig	Shows a list of the SmartWire-DT slaves contained in the currently stored target configuration.
swd_projectconfig	Shows a list of the SmartWire-DT slaves contained in the currently loaded project configuration.
swd_setnew	Stores the slaves currently found on the SmartWire-DT line as the target configuration.
swd_version [arg]	Shows version information for one / all SmartWire-DT slave(s).
swd_data [arg]	Shows the current status of the user data on the SmartWire-DT line.
swd_state	Shows status information of the SmartWire-DT network.
swd_diag [arg]	Shows extended diagnostics information of SmartWire-DT slaves.
swd_param [arg]	Shows extended parameter settings of SmartWire-DT slaves.
swd_umodules	Shows SmartWire-DT slaves replaced by universal modules.

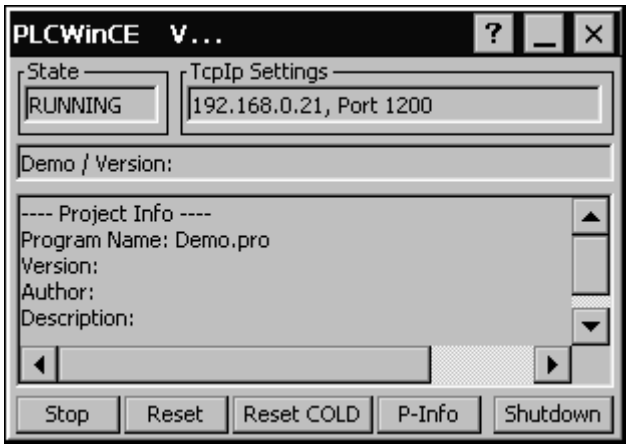
6 Operation

6.1 Startup behaviour

After startup the controller executes a system test. If there is no error detected, the controller switches to the operating state "stop" or "run". The system test contains the following tests:

- Memory test
- PLC-Program test

The result is visualized in the PLCWinCE window.



In addition the operating state of the controller depends of defined startup characteristics (➔ Chap. 6.3.1)

6.2 Switch off behaviour

With a voltage drop the program processing is immediately terminated and all necessary information for the restart is stored. After renewed switching on, the controller executes a restart.

6.3 Operating state of controller

Stop

The operating state "stop" is characterized by the following characteristics:

- A PLC program is located on the controller
- The PLC program is not executed

The operating state "stop" is assumed:

- After voltage switch on with defined startup characteristics "stop" (➔ Chap. 6.3.1)
- Via the PLC programming tool on the PC
- After a cycle time timeout / watchdog

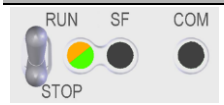
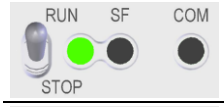
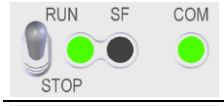
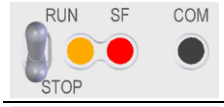
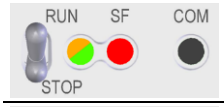
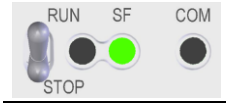
Run

In the operating state RUN the PLC program is executed cyclically.

The operating state RUN is assumed:

- After the power supply is switched on with defined "warm" start characteristics
- Via the PLC programming tool on the PC

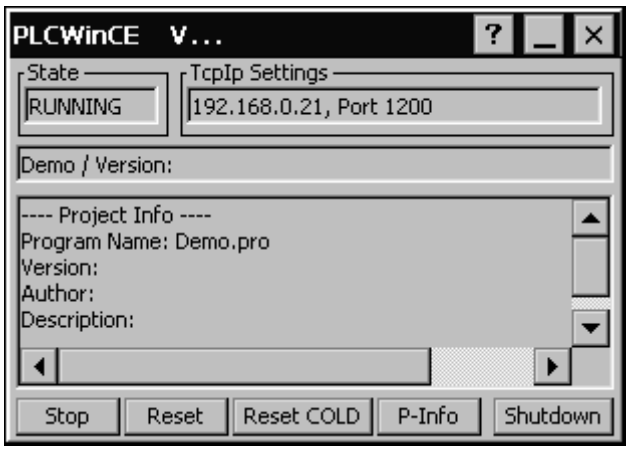
6.3.1 Overview operating- and error states

Operating -and error states	
	No PLC programm available … (RUN/STOP flashes orange/green)
	PLC is running …
	PLC is running and is communicating (CAN, Profibus) …
	PLC system fault …
	PLC system fault (unresolved Externals)
	PLC retain data fault Controller booted in detect-mode (pressed CTRL button)
	Controller booted in detect-mode (pressed CTRL button)

Please refer for more information on further operating- and error states the system description of Windows CE operating system.

6.4 Switching the operating state

The operating state is switched via the PLCWinCE window, via the PLC programming tool or the Mode switch (RUN/STOP).



It is also possible to define the startup behaviour after power supply switch on by means of the file **PlcRts.reg** in the directory **PlcRts.**

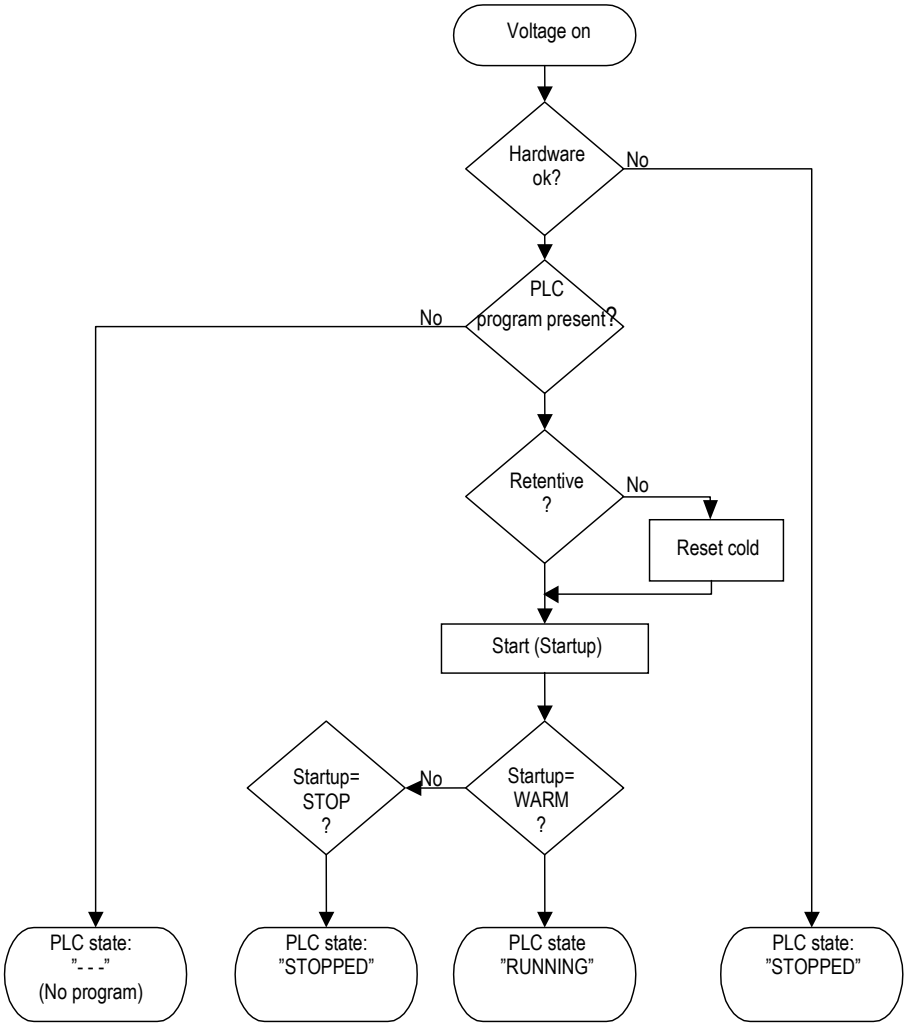
PlcRts.reg
[HKEY_LOCAL_MACHINE\SOFTWARE\Eaton\PLC Runtimesystem]
...
…
"Startup"="WARM"
…
…

The following settings are possible:

- Stop "Startup"="STOP"
- Warm start (Default setting) "Startup"="WARM"

6.5 Start behaviour

Procedure



Cold start

A cold start is initiated on the first start after loading the PLC program on the controller or after each reset cold. All variables of the PLC program are initialized with their startup values and the PLC program is started.

Warm start

With all further starts of the loaded PLC program or after each reset/reset warm takes place a warm start. The retentive (RETAIN, RETAIN PERSISITENT) variables keep their values, all remaining

variables are initialized with their startup values. Variables without explicit startup values are initialized with their standard startup values.

6.6 Stop behaviour

The processing of the PLC program is stopped at the end of the program cycle. Subsequently, a reset/reset warm takes place.

6.7 Reset behaviour

Reset / Reset warm

Corresponds to the initialization of warm start

Reset cold

This command resets all variables, also the retentive (RETAIN, RETAIN PERSISTENT) variables to the startup value.

Reset original

This command resets all variables, also the retentive (RETAIN, RETAIN PERSISTENT) variables to the startup value and deletes the PLC program on the controller. The controller is set back to the original state.

6.8 Test and startup

The controller supports following test and startup possibilities:

- Breakpoint/Single step mode
- Single cycle mode
- Force mode
- Online change

→ The power flow control is not implemented and is not supported.

Breakpoint / Single step mode

Breakpoints can be set in the PLC program. During the execution of an instruction tagged with a breakpoint, the PLC program is stopped task specific on the breakpoint. The subsequent instructions can be executed in single step mode. In this case the watchdog timing is deactivated.

→ The PLC program stops only on breakpoints of the task, which is defined in the task configuration as debug task.

Single cycle mode

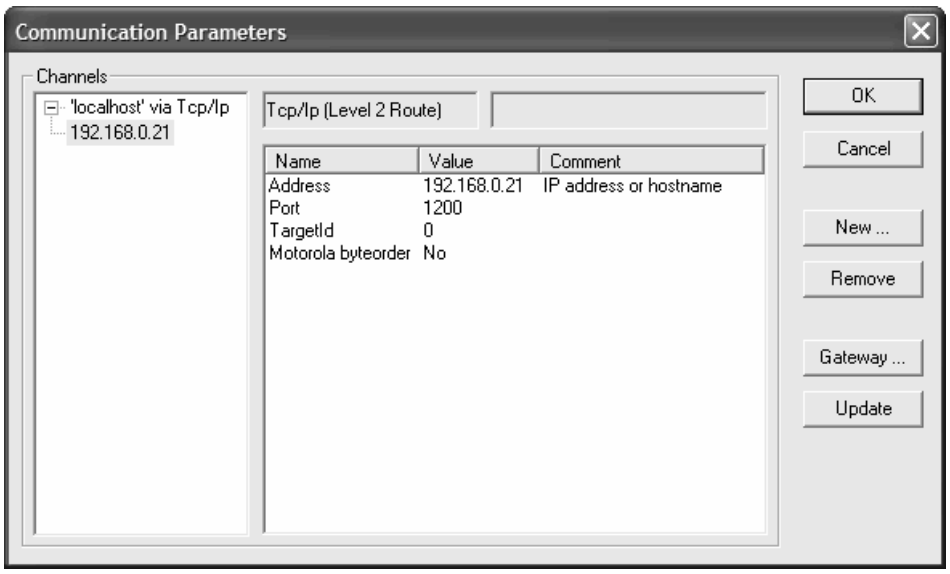
If single cycle mode is activated, only one task specific program cycle is executed. The outputs are enabled during the program cycle. At the end of the program cycle the output image is deleted. At the end of the program cycle the output image is deleted and the outputs are switched off. In this case the watchdog timing is deactivated.

Force mode

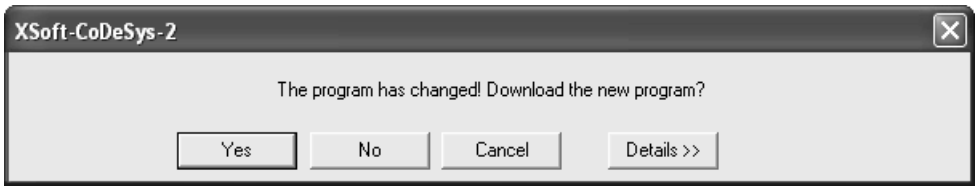
All variables of PLC program can be forced.

6.9 Program transfer

If the PLC program was compiled without errors in the PLC programming tool (PC), it can be downloaded to the main memory of the controller and can be started afterwards (Chap. 8)



During a program transfer from the PC to the controller, the program in the controller is compared with the program on the PC. If they are not identical, a prompt asks whether the program to be overwritten is. If this prompt is confirmed, then the controller is switched to the operating mode "stop" and the new program is loaded to the main memory.



Communication error

If no connection can be established between the PLC programming tool and the controller, the following points are to be checked:

- Physical connection
- Communication parameters in the PLC programming tool
- TCP/IP settings in the system settings on the programming PC
- TCP/IP settings in the system settings on the control

6.9.1 Create boot project

A PLC program is only power failure protected, if before voltage failure a boot project is created online and afterwards transferred to the controller.

→ Boot project can only be created "online".

7 Program execution and system time

7.1 Program execution

The operational sequence time of a PLC program is called a "task". In addition to the actual PLC program, all relevant system activities are processed as well.

For example, this includes the following system activities:

- Communication with the PLC programming tool
- Online changes
- Processing of CANopen protocol stack
- Processing of process images (IO-Update)

Program execution without task configuration

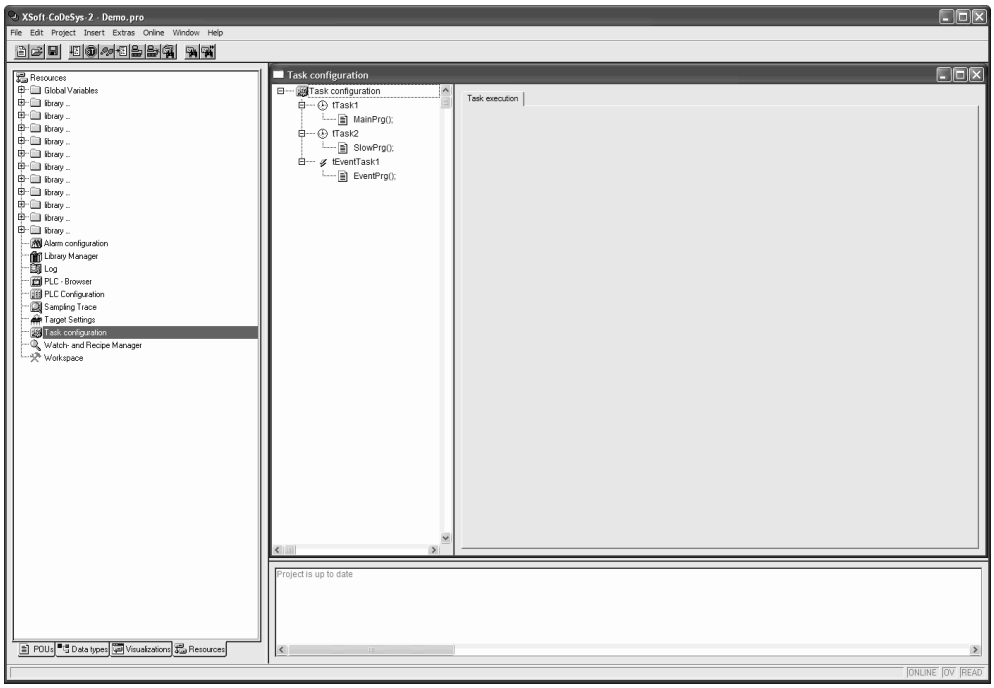
The first program unit, which is created in a new PLC program, carries automatically the name **PLC_PRG**. This starts program execution. The controller processes the PLC_PRG cyclically and free running with a minimum interval of 10 ms.

Program execution with task configuration

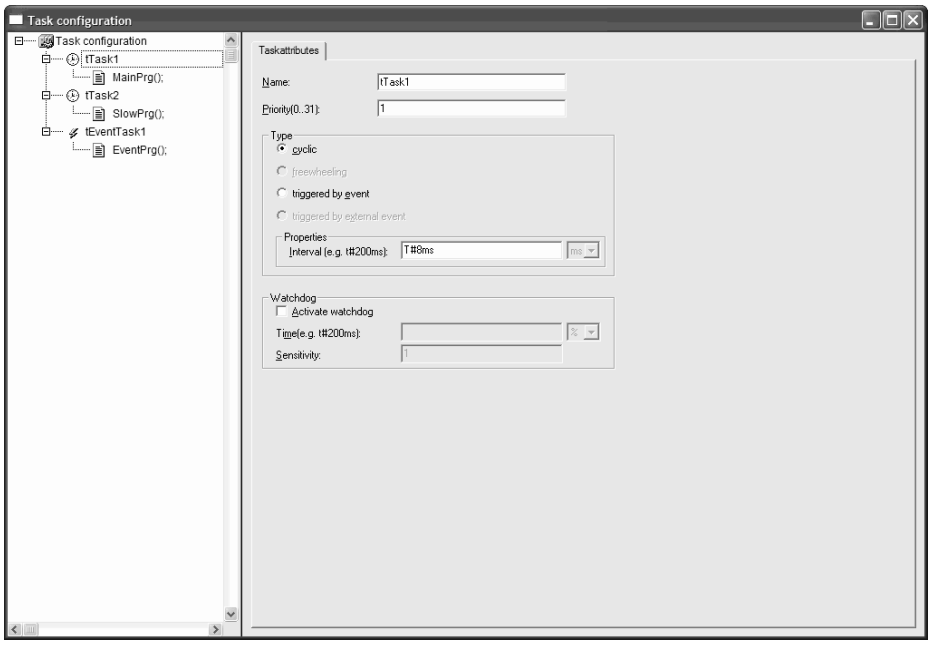
In addition to declaring the special program unit PLC_PRG, the processing of the PLC program can also be controlled via the task configuration. If a task configuration is defined, a program unit with the name PLC_PRG does not have to be created.

7.2 Task configuration

The task configuration is reached by via <Task configuration> in the 'Resources' tab.



Each task can be assigned a sequence by programs, which are to be executed by the call of the task. The task priority and task condition specify, in which timing sequence the tasks are to be processed. A PLC program can consist of several tasks of the same or a different priority, which are processed cyclically in parameterized time intervals or when particular events occur.



The correct definition of a task consists of following parameters:

- Task name
- Task priority
- Task type or task condition
- Task monitoring / Watchdog

Task name

Unique task name

Task priority

In order to prioritize the tasks, different task priorities can be assigned. The task priorities can be defined between 0 and 31. 0 corresponds of the highest and 31 of the lowest priority.

→ Tasks with same priority do not interrupt themselves mutually.

Task type or Task condition

The task condition can be either one time interval, after which the task is to be executed (cyclically), or a global variable, which causes their execution in the case of a rising edge (event-controlled).

An event-controlled task presupposes a cyclic task, which must be programmed in the program of the cyclic task. Each rising edge of the event initiates afterwards the execution program of the event-controlled task.

Task monitoring / Watchdog

Each task can be monitored with the help of the watchdog function (Chap. 7.4)

7.3 Multitasking

The PLC runtime system is a multitask system. This means that several tasks are processed at the same time. In a multitask system, individual tasks of different priority can interrupt each other.

The consequence of this is that the consistency of the input/output image is only ensured within the task which accesses the input/output image. If several tasks of different priority access the same input/outputs, consistency is only present with the task with the highest priority (➔ Chap. 7.9)

→ Always avoid the access of the physical input/outputs by several tasks in order to ensure a clear controller process.

7.4 Task monitoring / Watchdog timing

The task monitoring supervises the cyclic tasks of the PLC program. The task monitoring interrupts program processing, if the task exceeds a defined time in a defined frequency. By default the outputs of the controller are switched off when the watchdog time is triggered and the controller is switched into the operating state "stop". Afterwards the PLC program must be set back with "Reset".

The correct definition of a task monitoring consists of following parameters:

- Watchdog on/off
- Watchdog time
- Watchdog sensitivity

→ If the watchdog for the appropriate task is deactivated in the task configuration, no task monitoring will take place.

→ In a PLC program without a task configuration, the task monitoring is defined and activated with a watchdog time of 10ms by default.

Functionality

The watchdog is started at the beginning of each processing cycle and terminated after successful processing. The watchdog is triggered if the task duration is longer than the defined task interval.

The triggering of the watchdog depends further on watchdog sensitivity. The watchdog sensitivity specifies how many timeouts of the sequential task duration will cause the watchdog to be triggered.

The watchdog is triggered:

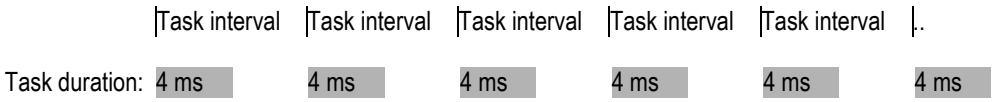
- with a watchdog sensitivity of 1, on the first timeout of the watchdog timing.
- with a watchdog sensitivity of x, only after there have been x timeouts in succession of the watchdog.
- The watchdog is also triggered by a continuous loop if the task duration is longer than the result from watchdog time * watchdog sensitivity. This criterion is used to detect and respond to continuous loops with pre selected watchdog functionality.

If you define a watchdog time of 10ms and the watchdog sensitivity of 5, the task is terminated at the latest after 10ms * 5 = 50ms.

The following examples should explain context of task interval, task duration and watchdog time.

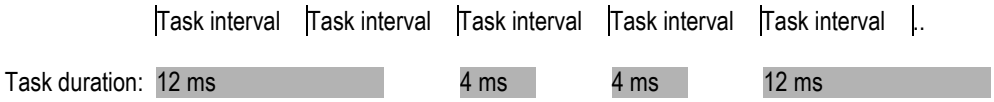
- defined task interval = 8ms
- defined watchdog time = 8ms
- defined sensitivity = 2

Example 1



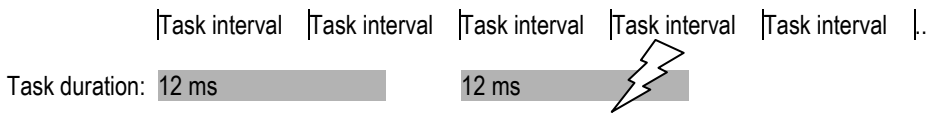
The watchdog timing is not triggered, because the task duration remains continually within the defined task interval.

Example 2



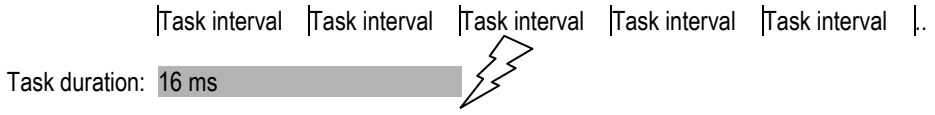
The watchdog timing is not triggered, because 2 timeouts of the sequential task duration do not occur in succession.

Example 3



The watchdog is triggered 8ms after the beginning of second task processing, because both task durations are longer than the indicated watchdog time and occur consecutively.

Example 4 (continuous loop)



The watchdog timing is triggered, because the task duration is longer than the multiplication of watchdog time and watchdog sensitivity.

7.5 Data retention

The controller has a storage area for retentive data (RETAIN, RETAIN PERSISTENT). This data is stored when switching the controller off.

If in certain circumstances a voltage drop terminates program processing in the middle of the program cycle, the data of the current program cycle will not be consistent.

With the next startup the controller is initialized with the data which was still written correctly by voltage drop.

Variable behaviour

Action	VAR	VAR RETAIN	VAR PERSISTENT	VAR RETAIN PERSISTENT
After Stop → Start	Initialization value	Value retained	Initialization value	Value retained
After warm start or reset / reset warm	Initialization value	Value retained	Initialization value	Value retained
After cold start or reset cold	Initialization value	Initialization value	Initialization value	Initialization value
After reset original	Initialization value	Initialization value	Initialization value	Initialization value
After 'Clear all' and subsequent program download	Initialization value	Initialization value	Value retained	Value retained

→ The variable behaviour of global variables corresponds also to the variable behaviour of local variables.

7.6 Direct pheriphery access

→ Direct periphery accesses (e.g. process image) are implemented for each specific target.

7.7 Interrupt processing

→ Direct interrupt processing is not supported.

7.8 System libraries, function blocks and functions

For designing the PLC program various function libraries are available.

→ Please refer to the detailed function descriptions in the relevant documentation of the function libraries.

7.9 Process image / IO-Update

During compilation of the PLC program the configuration of the inputs or outputs used in each task is attached to them.

Input image

An instruction "myVar := %IX0.0" leads to an entry in the configuration file, which describes that the value of the input %IX0.0 must be processed. If the byte %IB0 is used at the same time, the whole byte is processed instead of a bitwise access.

This configuration is created for each task and attached to the relevant task. Based on this configuration the inputs are read at the beginning of the task.

Output image

An instruction "%QX0.0 := myVar" leads in the configuration file to an entry, which describes that the value of the output %QX0.0 must be processed. If the byte %QB0 is used at the same time, the whole byte is processed instead of a bitwise access.

This configuration is created for each task and attached to the relevant task. Based on this configuration the outputs are read at the end of the task.

→ When creating the program concurrent accesses (i.e. access to the same input/output address from several tasks) should be avoided. This can lead to an inconsistent input/output image!

Warnings of concurrent access are displayed in the message window when compiling the PLC program.

The processing of the input/output image varies according to the different bus systems and the corresponding transmission mechanisms.

7.9.1 CAN-Bus

The inputs or outputs are packed in PDOs up to 8 byte data length. The Rx- and Tx-PDOs are updated in the task with the highest priority, which the PDO references.

The used inputs or Rx-PDOs are read at the beginning of the task and transferred to the input image.

The used outputs or Tx-PDOs are taken from the output image at the end of the task and written afterwards.

→ If the same Rx- or Tx-PDOs are used in different tasks, this can lead to cycle-inconsistent data:

With inputs it may occur that values are changed within a lower priority task cycle, because the inputs are read again by a task with higher priority.

Outputs can be sent too soon by another task, because the outputs are written again by a higher priority task.

7.9.2 Profibus

Profibus-Master CMM-MPI

The inputs and outputs are updated in an additional cyclic system task (driver task). This takes place asynchronously to the tasks of the PLC program.

The used inputs are read from the driver task at the beginning of the task and transferred to the input image.

The used outputs are taken from the output image at the end of the task, transferred to the driver task and written afterwards.

7.9.3 SmartWire-DT

The inputs and outputs are updated in a additional cyclic system task (driver task). This takes place asynchronously to the tasks of the PLC program.

The inputs will be transferred to the input image at the beginning of the first task in the PLC programs task configuration where they referenced.

The outputs are taken from the output image at the end of the first task in the PLC programs task configuration where they referenced.

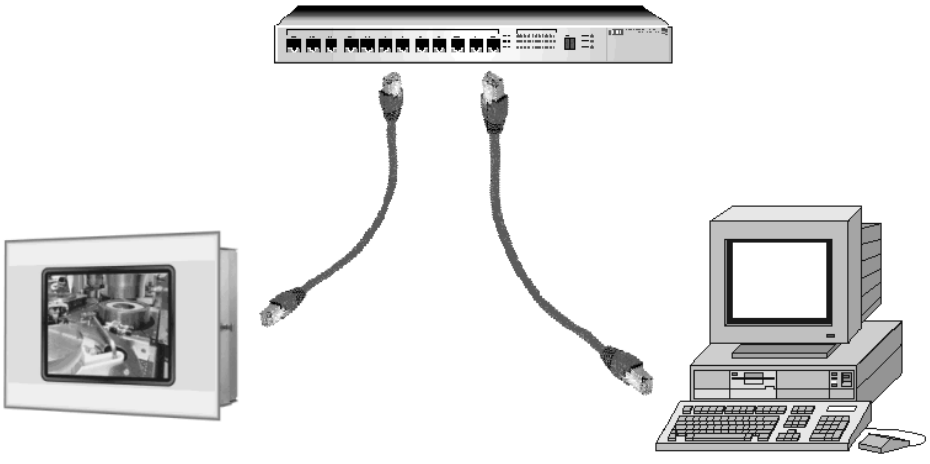
8 Connection establishment programming PC – Controller

8.1 Connection establishment with ethernet

Communication of the programming PC to the controller takes place with Ethernet and the TCP/IP protocol. For a direct connection, without Ethernet hub or switch, a crossed cable "Crossover" is to be used. On the controller a standard RJ45 jack is attached.

crossed RJ45 TwistedPair-Cable

For a connection to the PLC via an Ethernet hub or switch a straight (uncrossed) cable should be used.

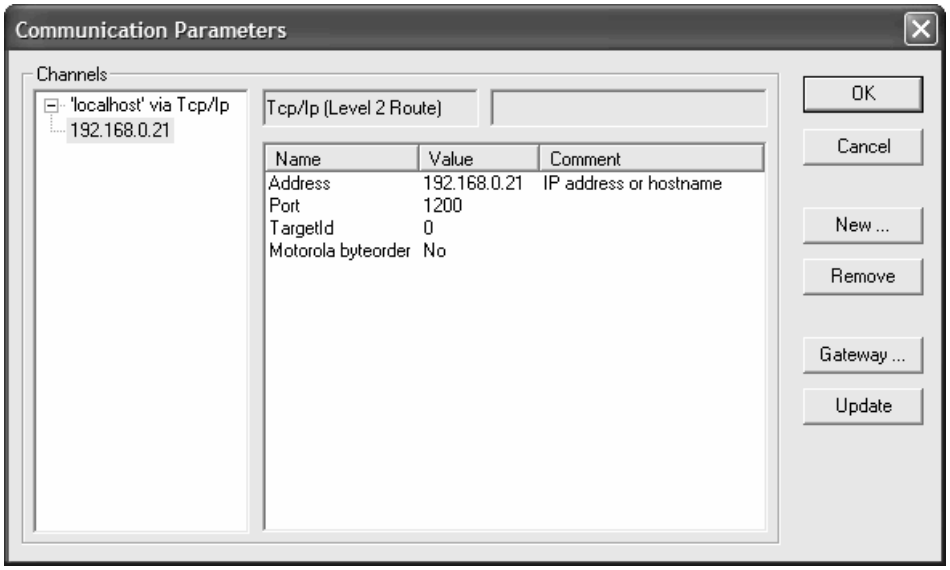


Straight RJ45 TwistedPair-Cable

Communication parameters

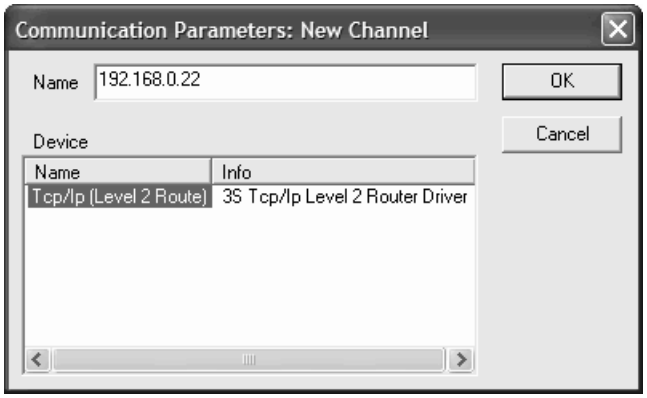
The 'Communication parameters' dialog is reached in the PLC programming tool via menu item <Online> <Communication parameters>. In this dialog the parameters are defined, which are valid for the communication between the programming PC and the controller.

All already created communication routes are listed in the 'Communication parameters' dialog under "Channels". Select now one of the channels, by clicking an entry with the mouse. The appropriate parameters are then indicated in the table. For a connection via Ethernet a channel of the type Tcp/Ip (level 2 route) should be used. The parameter address specifies the IP address of the controller which is used to communicate by this channel.



Setting a new channel

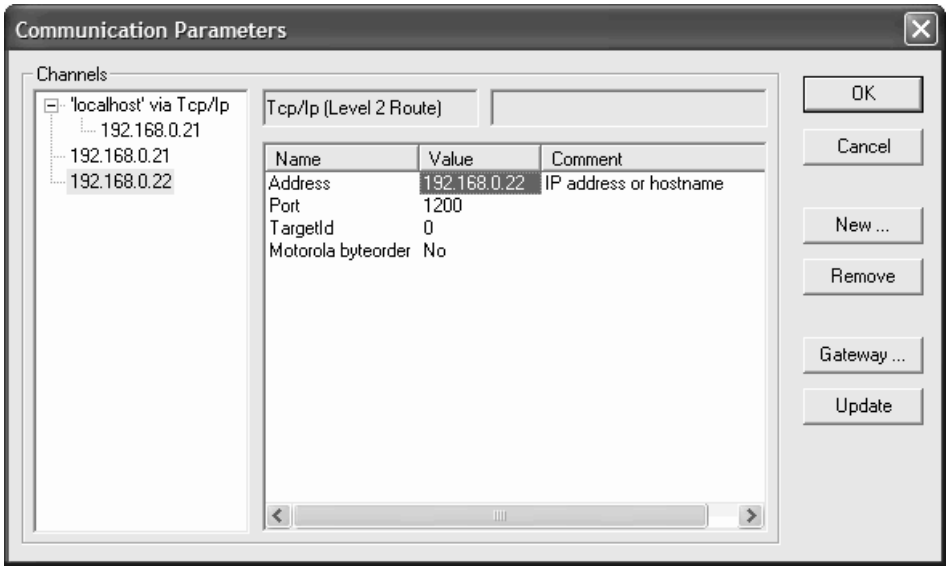
Click the button **New**. The 'Communication parameters: New channel' dialog appears:



The name used for the last entered channel is automatically proposed in the **Name** entry field. You can edit here the channel names. The channel names are solely for information, uniqueness is not mandatory but is recommended.

The available device drivers on the gateway computer are listed in the table under **Device**. In the **Name** column, click the driver **Tcp/Ip (Level 2 Route)** in order to select it. The corresponding comment, if any, appears in the **Info** column.

If you close the dialog 'New channel' with **OK**, the newly defined channel appears in the 'Communication Parameters' dialog as a new entry in **Channels** at the lowest position under the minus sign. So far, it is only stored locally in the project. At this point you can edit the **Value** column. Double click on the column value/row address and put the IP address of the controller and complete the entry with **Return**. Now confirm the entered parameters with **OK**, thus leaving the 'Communication Parameters' dialog.



The parameters for a channel already known by the gateway server can no longer be edited in the configuration dialog. The parameter fields appear grey. You can, however, delete the connection as long as it is not active.

Please note that the deletion of a channel is not reversible. It occurs at the moment that you press on the button **Remove** !

→ Save your PLC program with the new communication parameters, compile this and afterwards log on the controller.

→ Make sure that the IP addresses of the programming PC and the controller belong to the same address family. Please refer to detailed information in the documentation "Networks in brief".

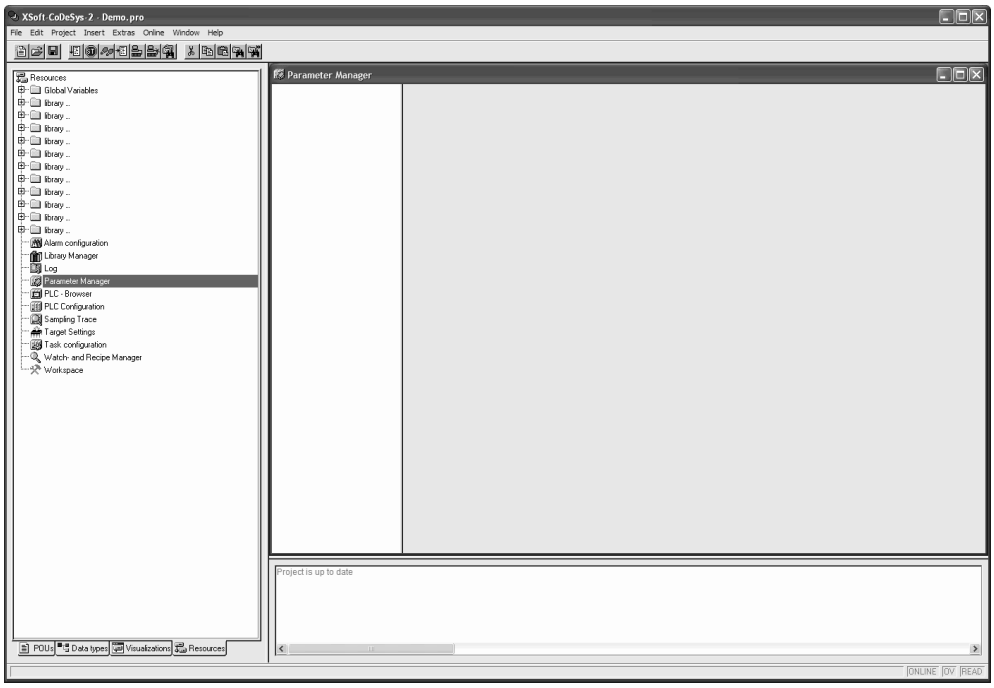
8.2 Scan/Modify the IP address

The "Remote Client" is available for modifying and scanning the IP address.

→ Please take information from the system description of the operating system Windows CE, chapter "Connecting with XC-150".

9 Parameter manager / Object directory

If the option is activated, the entry **Parameter Manager** appears in the 'Resources' tab. This makes it possible to create an object directory for variables and parameters, which serve a purposeful active data exchange with other controllers.



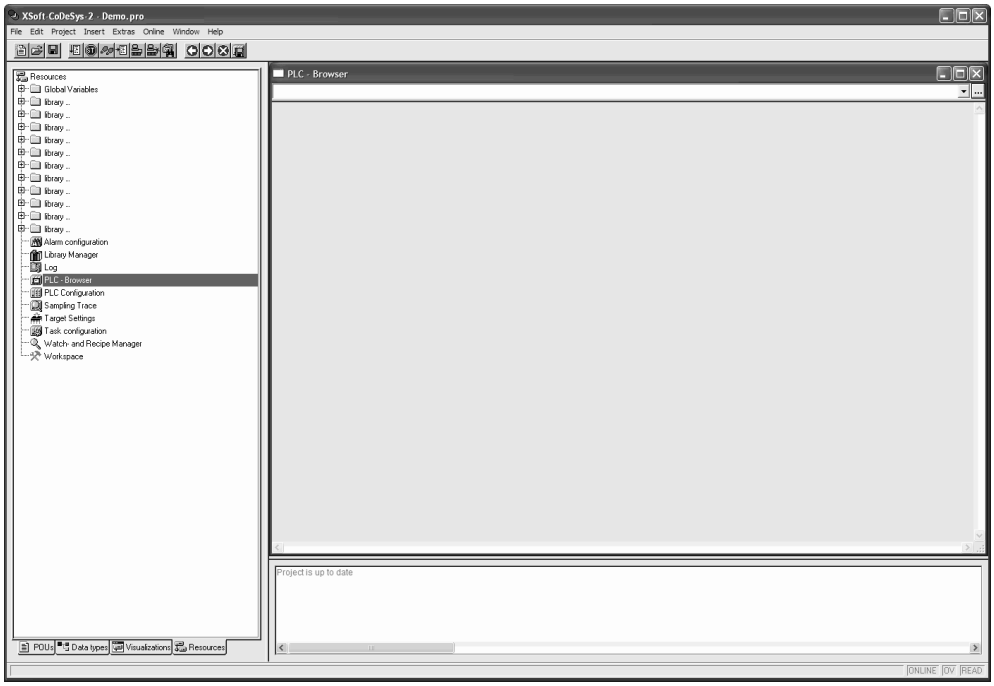
→ The object directory functionality is supported in connection with an inserted CAN-Device in the PLC configuration.

→ Please refer to detailed information in the CoDeSys V2.3 user manual or the online help of the PLC programming tool.

10 PLC browser

The PLC browser is a text-based controller monitor (terminal). Commands for the request of specific information from the controller are entered in an entry line and sent as string to the controller. The returned response string is displayed in a result window of the PLC browser. This functionality is used for diagnostics and debugging tasks.

The PLC browser is reached via <PLC-Browser> in 'Resources' tab.



The available commands were divided into two groups:

- Standard browser commands
- Target system specific browser commands

These commands are managed in a file and implemented accordingly in the PLC runtime system.

Command	Description
?	Gets the list of implemented and available commands
...	...

In order to use the PLC browser functionality, you must be connected online with the target system.

11 Alarm configuration

All PLC target systems with enabled target visualization support the CoDeSys alarm configuration (➔ Chap. 4.5).

Please refer to detailed information in the CoDeSys V2.3 user manual, the CoDeSys visualization user manual or the online help of the PLC programming tool.

12 Connecting to visualization / Generating of the symbol file

The symbol file provides a basis for communication between the controller and a possible used visualization. The content of this symbol file is configured in the PLC programming tool. During compilation this symbol file is generated and loaded with the subsequent program download to the controller.

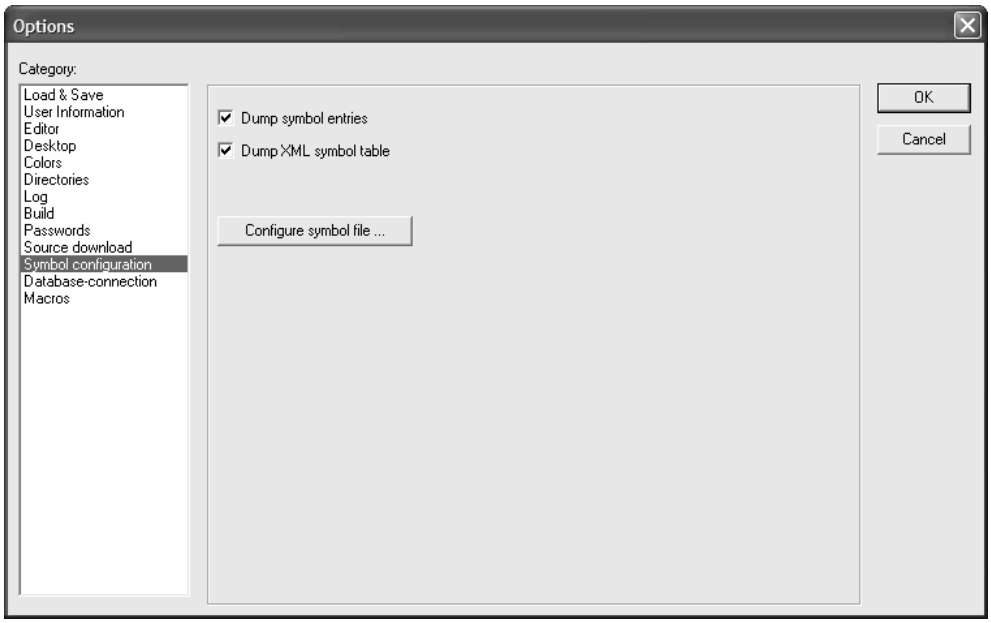
12.1 Configure symbol file

The content of the symbol file is configured in the symbol configuration. The symbol configuration is reached by menu item <Project> <Options>.

→ If the simulation is activated in the 'Online' menu item, the functionality 'symbol configuration' does not appear in the 'Options' dialog.

To generate the symbol file during compilation, the option **Dump symbol entries** must be enabled.

Visualizations support partially the import of the symbol file in XML format. To generate the symbol file during compilation, the option **Dump XML symbol table** must be enabled.

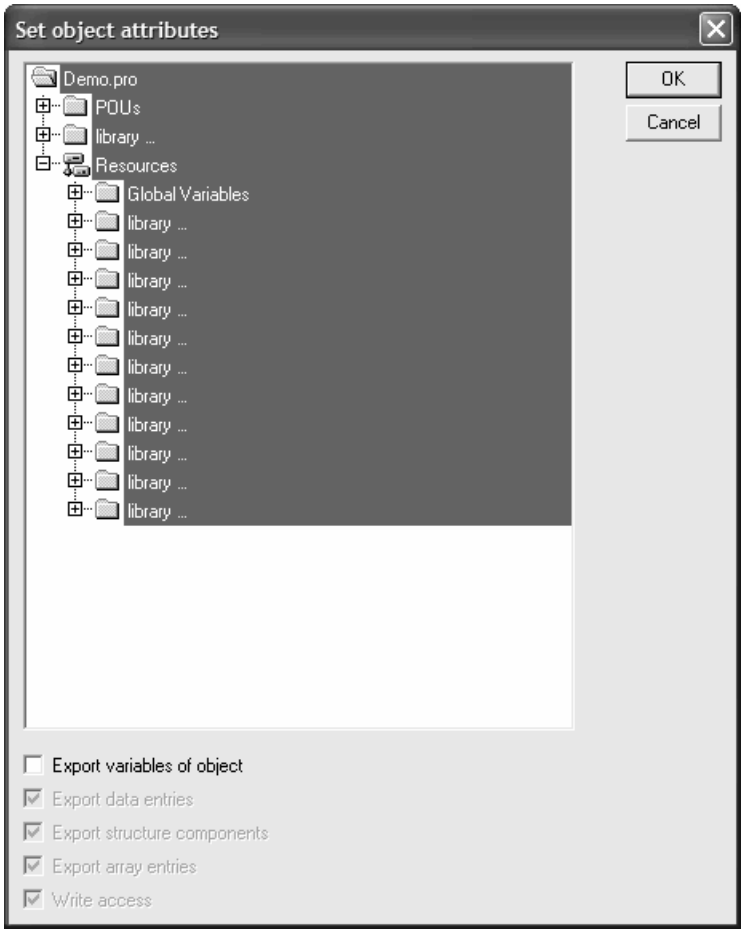


The **Configure symbol file** button is used to define from which objects the variables are exported into the symbol file.

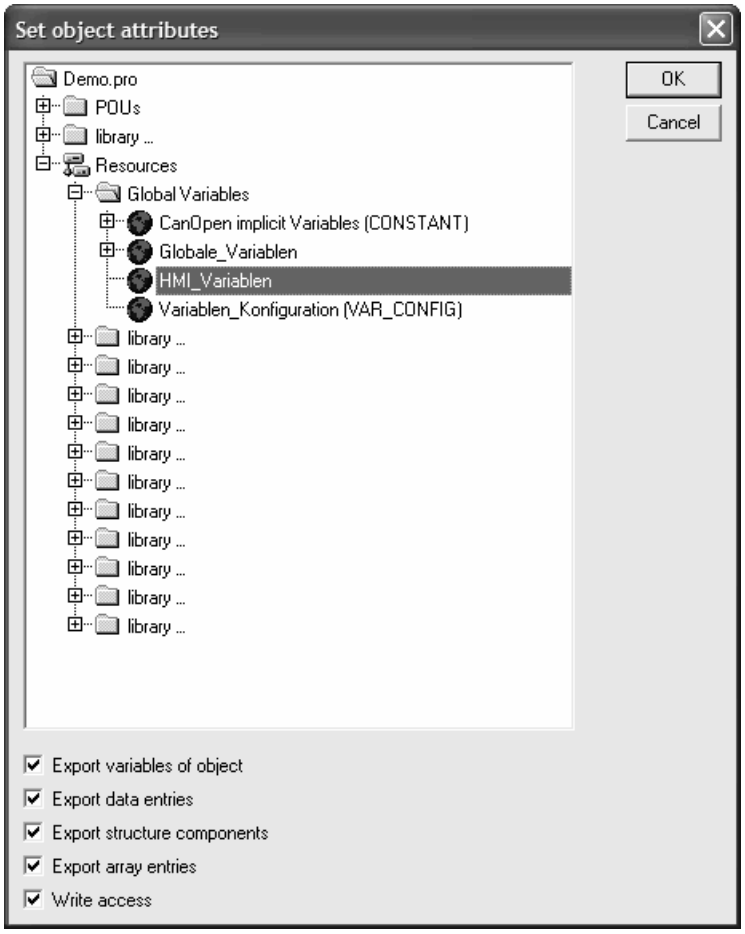
→ It is recommended that only the variables needed for the variable exchange with visualization are exported into the symbol file. Therefore the range of these variables should be structured. This is done by defining global variable sheets or by using of pragma-instructions.

→ Please refer to the detailed information in the CoDeSys V2.3 user manual or the online help of the PLC programming tool.

All objects are selected by default. When first opening this dialog, deactivate the option 'Export variables of object' for all objects.

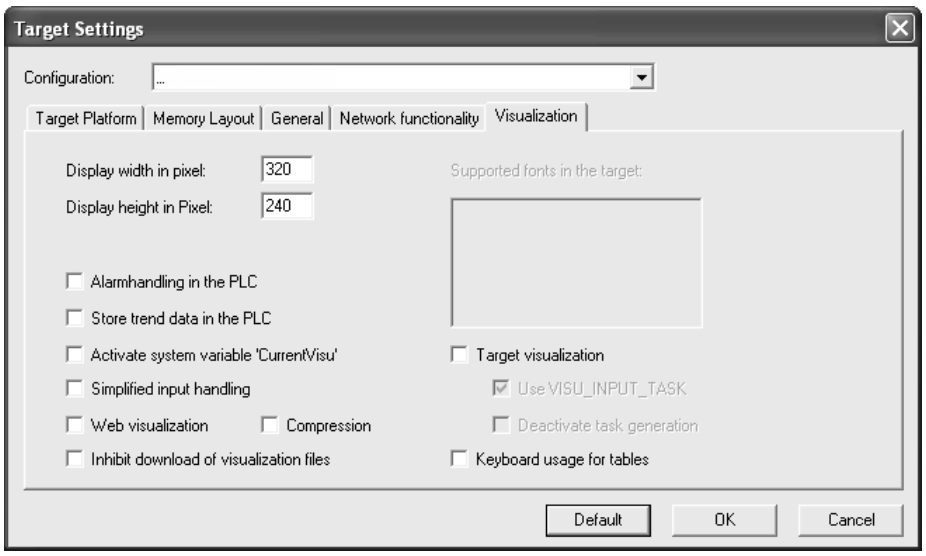


Now activate selectively the 'Export variables of object' option for the desired objects.



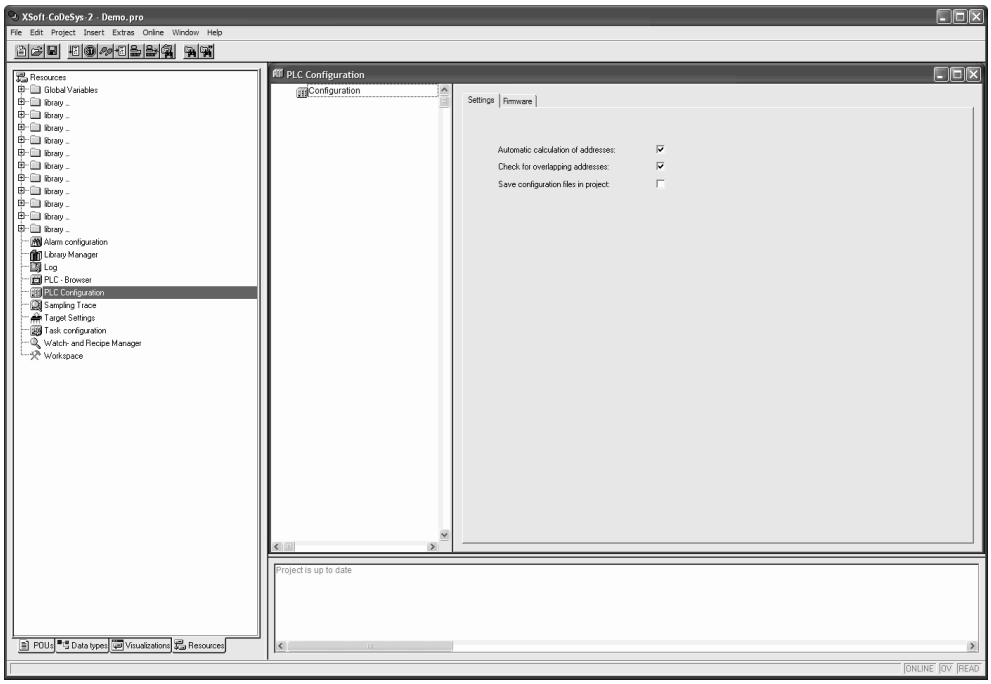
12.2 Download symbol file

In order to load the symbol file is loaded onto the controller during project download, the **Download symbol file** option must be activated in the 'General' tab of the target settings



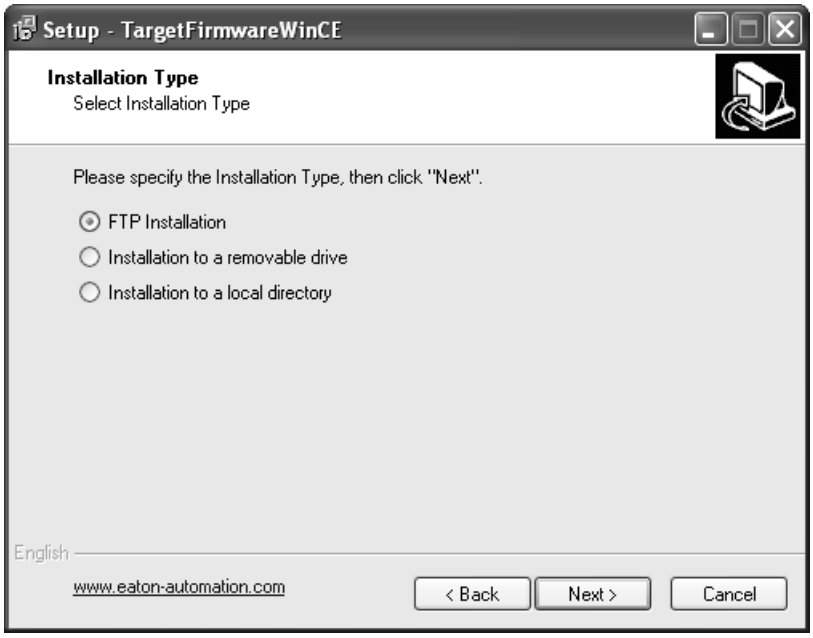
13 Target system installation and firmware update

The target system installation or the firmware update is implemented by calling the "TargetFirmwareWinCE" program and is reached in the PLC programming tool via <PLC configuration> in the 'Firmware' tab.



Activate in the area "Update operating system" the button **Start** und select appropriate Firmware.

The following installation types are available:



FTP Installation : The installation of the directories **PlcRts** and **PlcPrg** is made by FTP.

Installation to a removable drive : The removable drive (e.g. CompactFlash™) must be available on the programming PC by an adapter (e.g. PC Card adapter). The installation of the directories **PlcRts** and **PlcPrg** takes place directly on the removable drive.

Installation to a local directory : The installation of the directories **PlcRts** and **PlcPrg** takes place in a local directory on the programming PC. Subsequently, the directories **PlcRts** and **PlcPrg** must be copied manually into the root directory on removable drive (e.g. CompactFlash™).

→ The stored PLC program remains by firmware update.

→ The listings PlcRts and PlcPrg are not deleted with a target system installation or firmware update. Existing files are only overwritten.

14 Licensing

14.1 PLC programing tool

The PLC programming tool is subject to license.

➔ If no series number or license key is available by the installation of the PLC programming tool, the target systems are installed in the demo mode.

14.2 PLC runtime system

The PLC runtime system is license requiring and needs **100 license points** on the PLC target.

➔ On the controller, a PLC runtime system is already installed. For a necessary update of the operating and/or PLC runtime system see chapter "Target system installation and firmware update" (➔ Chap. **Fehler! Verweisquelle konnte nicht gefunden werden.**).

➔ If license points are missing with the start of the PLC runtime system, the processing of the PLC program is not started. The PLC program starts in the operating state "stop".

The processing of the PLC program can be made afterwards in the PLC programming tool via menu item <Online> <Start>.

14.3 Target visualization

The Target visualization is license requiring and needs **100 license points** on the PLC target.

➔ If license points are missing with the start of the PLC runtime system, the processing of the PLC program is not started. The PLC program starts in the operating state "stop".

The processing of the PLC program can be made afterwards in the PLC programming tool via menu item <Online> <Start>.

14.4 Web visualization

The Web visualization is not license requiring and needs **no license points** on the PLC target.

15 Alteration index

Index	Date / Visa	Changes
01	13.07.12 / DL	Initial version
02	31.10.12 / DL	Adaptions and corrections: Chapter SmartWire-DT configuration, Overview operating- and error states
03	19.06.13 / DL	Adaptions and corrections: Chapter operating state of controller, Overview operating- and error states extended
04	12.05.14 / DL	Adaptions and corrections due XSOFT-CODESYS V2.3.9 SP4, Chapter "Target system installation and firmware update" changed